Lecture 1 : Tuesday 5/9

The first part of the course is an introduction to the subject of *Enumerative Combinatorics*. While the second of these words is a bit hard to explain, as it refers to a rather broad spectrum of mathematical problems and techniques, the first clearly suggests that the subject has something to do with counting/enumeration. Actually, the general set-up is the following :

You have a function $f : \mathbf{N} \to \mathbf{N}$ which you know a bit about. More precisely

(i) you have a precise, concrete description of the function as ”counting” some collection of objects
(ii) usually, you know intuitively that $f(n)$ is a “fast-growing” function of n.

This is a bit informal, but the point is the following : since you have a pretty concrete description of what the function is counting, you could in principle evaluate $f(n)$ by actually performing the count. But (ii) implies that for all practical purposes, this is not feasible. Hence, what you need is some more intelligent way of evaluating, or at least estimating $f(n)$.

The following simple example illustrates many of the basic features of the type of problem with which we will be concerned :

Example 1 : How may subsets are there of the set $\{1, 2, 3, ..., 2006\}$. Alternatively, how many 2006-digit binary words are there ?

The first thing one should ask oneself here is why the two questions have the same answer. Well, that's because to any subset of $\{1, 2, ..., 2006\}$ we can associate a 2006-bit word in the following way : if the subset contains 1, then the first bit is a 1, and is 0 otherwise. If the subset contains 2, then the second bit is a 1, and 0 otherwise. And so on ....

It should be clear that this establishes a *one-to-one correspondence* (mathematicians also use the word *bijection*) between the subsets of $\{1, 2, ..., 2006\}$ and 2006-bit words. Actually, this is a common feature of enumerative problems : that there may be several different “concrete” ways of describing the problem, which are all equivalent to one another, but whose equivalence may not necessarily be obvious. In fact, sometimes finding a clever way to enumerate depends on finding just the right concrete description of the problem.

The next step is to explain how this example fits into the general set-up. The thing to notice here is that there is no reason to expect there is anything special about the choice of 2006. I could choose any natural number I want and ask the same question. I expect intuitively that, if there is a clever way to enumerate the subsets/binary words, then it should work for any choice of set size/word length.

So the function $f : \mathbf{N} \rightarrow \mathbf{N}$ which we are interested in here is described explicitly by

Alt 1 : $f(n) =$ number of subsets of $\{1, 2, ..., n\}$.
Alt 2 : $f(n) =$ number of n-bit binary words.

So how to compute $f(n)$? A good strategy is often to work out the first few values and write out the full list of corresponding subsets/words in the hope of finding a pattern. This works here !

First note that $f(1) = 2$ since if you have only one bit, then there are two choices for it, either 0 or 1.

Next, $f(2) = 4$ since there are four 2-bit words, namely 00, 01, 10, 11.

Let's go one step further. We have $f(3) = 8$ since there are the following eight 3-bit words : 000,001,010,011,100,101,110,111.

So what comes next after 2,4,8,... ? A reasonable guess is 16, i.e.: that we double the number of words each time. This turns out to be true, i.e.: we have that

$$f(n + 1) = 2 \cdot f(n) \quad \forall \, n \geq 1. \tag{1}$$

The proof is easy : If I want to write down an $(n + 1)$-bit word, I can first write down an n-bit word, and then tag on either a 0 or a 1. In other words, every n-bit word gives rise to exactly two $(n + 1)$-bit words.

Eq.(1) is an example of a so-called *recurrence relation* : it defines a recursive procedure for computing the values $f(n)$. In order to carry out the computation, on a computer say, we also need a starting value, or so-called *initial condition.* And we know that

$$f(1) = 2. \tag{2}$$

From (1) and (2) we can compute all the values of $f(n)$. But, even better, we can write down a simple explicit formula for our function, namely

$$f(n) = 2^n. \tag{3}$$

Not only that, but this formula is sufficiently simple that it is easy to estimate the *rate of growth* of $f(n)$. For example

$$f(n) = 2^{2006} = 10^{2006 \cdot \log_{10} 2} = 10^{603.866...}$$

hence a 604-digit decimal number.

To summarise, for this example our solution is about as satisfactory as it can get : we get an explicit formula for $f(n)$ which gives clear information about the asymptotic behaviour of the function. Things aren't always so easy, as we shall see !

Multiplication principle

The above example in particular illustrates a simple but very general counting principle which has many applications.

Proposition 1 (Multiplication Principle) *If we have n_1 balls of color C_1, n_2 balls of color C_2,...., n_k balls of color C_k, then the number of ways of choosing k balls, one of each color, is $\prod_{i=1}^{k} n_i$.*

Don't confuse this (many students do !) with

Proposition 2 (Addition Principle) *Under the same hypotheses as above, the number of ways of choosing a single ball, in any color, is $\sum_{i=1}^{k} n_i$.*

Actually, as we shall see in due course, there are many situations where one has to apply both principles (hence the reason for people getting confused !). For the moment, though, only MP is of interest.

Example 2 : Give an upper bound on the number of cars in Sweden.

SOLUTION : Every car has a unique licence plate, so it suffices to upper bound the number of possible licence plates. A simple bound is given by the total number of possible combinations xxx-ddd, where xxx is a 3-letter combination, each letter being chosen freely from among A-Z, and ddd is a 3-digit combination, each digit being chosen freely from 0-9.

There are 26 possible choices for each letter, 10 possible choices for each digit. By MP, the number of possible combinations is $26 \cdot 26 \cdot 26 \cdot 10 \cdot 10 \cdot 10 = 26^3 \cdot 1000 = 17,576,000$. I guess there aren't that many motor vechicles in Sweden (yet)!

We now explore some special applications of the multiplication principle.

NOTATION/TERMINOLOGY : Let n, k be two positive integers with $k \leq n$. Suppose we have n balls in n different colors. We want to choose k of them and place them in a row : in other words, if we choose the balls one by one, then the order in which we choose them is significant. We denote by $P(n, k)$ the number of ways of performing this choice.

If instead we are not interested in the order in which the balls are chosen, but just which ones, then the number of ways of making this choice is denoted $C(n, k)$, or alternatively $\left(\begin{array}{c} n \\ k \end{array} \right)$.

The first method of choosing balls is called *ordered choice without repitition/replacement*, the second is called (naturally !) *unordered choice without repitition/replacement*. The "without replacement" bit refers to the fact that you can't choose the same ball twice.

Proposition 3

$$P(n, k) = \frac{n!}{(n-k)!} \tag{4}$$

and

$$C(n, k) = \frac{P(n, k)}{k!} = \frac{n!}{k! \cdot (n-k)!}. \tag{5}$$

PROOF : (i) Think of choosing the balls one by one. There are n choices for the first ball. Once that's been removed, there are $n-1$ choices for the next one, then $n-2$ choices for the one after that and so on. Hence, by MP, we obtain that $P(n, k) = n \cdot (n-1) \cdot (n-2) \cdots (n-k+1)$. Multiplying above and below by $(n-k)!$ gives (4).

(ii) Once k balls have been chosen, there remain $P(k,k) = k!$ possible ways to order them. In other words, every unordered choice of k balls gives rise to $k!$ different ordered choices of the same balls. This means precisely that $P(n,k) = k! \cdot C(n,k)$, which implies (5).

NOTE : In the special case $n = k$ we have $P(n,n) = n!$. Here we are just counting the number of ways of ordering/sorting all n balls. An ordering/sorting of a bunch of objects is called a *permutation*. Hence (4) implies that there are $n!$ possible permutations of n objects.

EXAMPLE 3 : Eight guys compete in the Olympic 100-metre final. How many possibilities are there for the medal winners if (i) we don't care who wins which medal (ii) we do care ?

SOLUTION : (i) $C(8,3) = 56$ (ii) $P(8,3) = 336$.

Lecture 2 : 8/9

Remark : The formulae (4) and (5) for $P(n,k)$ and $C(n,k)$ both involve the factorial function. For these formulae to be really satisfactory from the point of view of efficient computation, we would like an efficient method for estimating the size of the (super-exponentially fast growing) factorial function. Such an estimate is provided by *Stirling's Formula.* First some notation : if $f(x)$ and $g(x)$ are functions of a real variable x, then one writes $f(x) \sim g(x)$ to mean that $\lim_{x\to\infty} f(x)/g(x) = 1$. Now we can nicely formulate the result :

Theorem 4 (Stirling's Formula)

$$n! \sim n^n e^{-n}\sqrt{2\pi n}. \tag{6}$$

Proof : Standard proof requires complex analysis and hence is beyond the scope of this course. Note that (6) does indeed give a very quick means to estimate $n!$, namely it reduces the problem to a computation with logarithms.

Our next task is to prove a number of combinatorial properties of the numbers $C(n,k)$:

Proposition 5 *The numbers $C(n,k)$ satisfy the following identities :*
(i)

$$C(n,0) = 1, \quad C(n,1) = n \quad \textit{for any } n. \tag{7}$$

(ii)

$$C(n,k) = C(n,n-k). \tag{8}$$

(iii)

$$C(n,k) = C(n-1,k) + C(n-1,k-1). \tag{9}$$

Note : Eq. (9) is called *Pascal's identity.*

Proof : All these identities could easily be proven 'algebraically' by just using the formula (5). However, one doesn't get much insight that way into where the identities come from, for which instead one must argue combinatorially. This we now do :

(i) $C(n,0)$ is the number of ways to choose no balls from some collection of

n balls, and obviously there is one way to do this no matter how many balls you have to choose from. Similarly, $C(n, 1)$ is the number of ways to choose one ball from n, and obviously there are n ways to do this.

(ii) $C(n, k)$ is the number of ways to choose k balls from n. Each choice of k balls corresponds to a rejection of $n - k$ balls. Hence $C(n, k)$ is the number of ways to reject $n-k$ balls from n. And this must, of course, equal the number of ways to instead choose $n-k$ balls from n, namely $C(n, n-k)$.

(iii) Isolate one of the n balls, call it B. When one chooses k balls from n, two possibilities arise : either ball B is chosen or it is not. If B is chosen, then it remains to choose $k-1$ balls from $n-1$, so there are $C(n-1, k-1)$ ways to choose the remaining balls. If ball B is not chosen, then it still remains to choose k balls, but only from among $n - 1$ of them, so there are $C(n - 1, k)$ ways to perform the choice. By the AP, there are thus in all $C(n-1, k-1)+C(n-1, k)$ ways to choose k balls from n, which proves (9).

The numbers $C(n, k)$ can be presented as a triangle, where n increases downwards and k from left to right. For obvious reasons, this triangle is called *Pascal's triangle*. For a much nicer drawing of it than anything I could concoct here, see for example

http://www.math.umass.edu/ mconnors/fractal/generate/pascal2.gif

Notice that each number in the triangle is the sum of the two numbers directly above it (this even applies at the edges if we think of there being zeroes outside the triangle). This is just Pascal's identity.

The numbers $\left(\begin{array}{c} n \\ k \end{array} \right)$ are sometimes called *binomial coefficients*. The reason is their appearance in the so-called *Binomial Theorem* from algebra, which says how to expand a power of a sum of two indeterminates :

Theorem 6 (Binomial Theorem) *Given indeterminates x and y and a non-negative integer n we have that*

$$(x+y)^n = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) x^k y^{n-k}. \tag{10}$$

PROOF : Think of what one does when one actually expands $(x+y)^n$. One has a product of n identical factors

$$(x+y) \times (x+y) \times \cdots \times (x+y) \quad (n \text{ times}).$$

Because of the distributive law, the result of the expansion is a sum of terms obtained by choosing either x or y from each factor in all possible ways and multiplying them together. (Note that, by MP, this will result in a total of 2^n terms to be added together). Clearly, each individual term is of the form $x^k y^{n-k}$ for some k with $0 \leq k \leq n$. The question is, how many times does such a term appear for a fixed k ? Well, from what we've just said, it follows that the number of times such a term will appear equals the number of ways of choosing k of the n factors from which one picks x instead of y. But, by definition, this is just $\left(\begin{array}{c} n \\ k \end{array} \right)$, and the theorem is proved.

Example 4 : For any $n \geq 0$ we have

$$2^n = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right). \tag{11}$$

There are many ways to understand this. I more or less showed it in the last proof : by MP, there are a total of 2^n terms resulting from the expansion of $(x+y)^n$ and for any k with $0 \leq k \leq n$, the number of these terms which have the form $x^k y^{n-k}$ is just $\left(\begin{array}{c} n \\ k \end{array} \right)$. A particularly neat way of formulating this argument is to substitute $x = y = 1$ in (10).

If one thinks instead in terms of n-bit binary strings, for example (see last lecture), then $\left(\begin{array}{c} n \\ k \end{array} \right)$ is the number of such strings containing exactly k zeroes, since once one chooses the positions of the k zeroes the whole string is uniquely determined.

Exercise : Insert $x = 2, y = 1$ alternatively $x = 1, y = -1$ in (10), and interpret the resulting identities combinatorially (in terms of binary strings, for example).

A generalisation of the Binomial Theorem is the *Multinomial Theorem* which, as it name suggests, tells one the result of expanding a power of a sum of an arbitrary number of indeterminates.

Notation : Let n be a non-negative integer and let $(n_1, ..., n_r)$ be an ordered r-tuple of non-negative integers with $n_1 + \cdots + n_r = n$. We denote

$$C(n : n_1, ..., n_r) := \left(\begin{array}{c} n \\ n_1, ..., n_r \end{array} \right) := \frac{n!}{\prod_{i=1}^{r} n_i!}.$$

The numbers $C(n : n_1, ..., n_r)$ are called *multinomial coefficients*, for the following reason :

Theorem 7 (Multinomial Theorem) *Given indeterminates $x_1, ..., x_r$ and a non-negative integer n, we have that*

$$(x_1 + \cdots + x_r)^n = \sum_{(n_1,...,n_r): n_i \geq 0 \ and \ \sum n_i = n} \left(\begin{array}{c} n \\ n_1, ..., n_r \end{array} \right) x_1^{n_1} x_2^{n_2} \cdots x_r^{n_r}. \tag{12}$$

Proof : To be more explicit, the sum in (12) is taken over all ordered r-tuples of non-negative integers which sum to n. Observe that in the case $r = 2$ we recover the Binomial Theorem. Instead of writing out a proof of (12) here, we leave it as an exercise to the reader to deduce it from the following more concrete combinatorial interpretation of the multinomial coefficients :

Proposition 8 *Suppose we have n balls in r different colors, with n_i balls in the i:th color. If we don't distingush between balls of the same color, then the number of possible permutations of the n balls is $C(n : n_1, ..., n_r)$.*

Proof : If all the balls were considered as distinguishable objects, then there would of course be $n!$ possible ways to permute them. Consider any such permutation. If balls of the same color are now considered indistinguishable, it means we can permute balls of the same color amongst themselves without affecting the overall arrangment of the n balls. There are $n_i!$ ways to permute the n_i balls of color i amongst themselves. Hence, by MP, there are $\prod_{i=1}^{r} n_i!$ inter-color permutations which don't affect the overall arrangement of the n balls. It follows that we must divide $n!$ by this number to get the number of possible mutually distinguishable arrangments of the n balls, Q.E.D.

The last topic for today is *unordered selction with repitition allowed.* The set-up is similar to before : we have n balls and wish to count the number of possible ways to choose k balls, where the order of choice is unimportant. The difference is that, this time, it is allowed to choose the same ball as often as one likes. In particular, this means that k can be greater than n. In fact, it is simpler to think of their being an unlimited supply of balls, but in n different colors, and of balls in the same color being indistinguishable.

WARNING ! (NOTATIONAL NIGHTMARE) There is no standard notation, as far as I'm aware, for the number of ways of performing unordered choice with repitition. Not only that, but it is standard procedure to reverse the roles of n and k. Hence one talks of their being k different colors and one wants to make a choice of n balls. Once you've got your head around this, you'll be ready for the main result :

Proposition 9 *The number of ways to choose n balls from an unlimited supply of balls in k different colors, where the order of choice is unimportant and balls in the same color are considered indistinguishable, is just* $\begin{pmatrix} n+k-1 \\ k-1 \end{pmatrix}$.

PROOF : The following ingenious idea I like to think of as the '*dots and dashes method'*. Consider a collection of $n+k-1$ symbols, of which n are dots and $k-1$ are dashes. The dots are considered indistinguishable from one another, as are the dashes. Then $\begin{pmatrix} n+k-1 \\ k-1 \end{pmatrix}$ is just the number of ways of arranging these symbols in a line, as the only choice one has to make is which $k-1$ of the $n+k-1$ positions will hold dahses.

Now the idea is that there is a natural 1-1 correspondence between all these possible arrangements of dots and dashes and all possible ways of choosing n balls according to the present rules. Since balls of the same color are indistinguishable, all that matters is how many balls of each color are chosen. Given an arrangement of dots and dashes, we can interpret the number of dots appearing before the first dash as the number of balls chosen in color 1. Then the number of dots between the first and second dashes is interpreted as the number of balls chosen in color 2. And so on, with the number of dots after the last $((k-1)$:st) dash representing the number of balls chosen in color k.

It is pretty clear that this gives a 1-1 correspondence between all possible ways of choosing the balls, and all possible arrangements of dots and dashes. Hence the proposition is proved.

Example 5 : Let's suppose there are 15 political parties taking part in the Swedish parliamentary election (I don't know the exact number). Let's also suppose exactly 5 million people vote. How many possible outcomes are there, in terms of the number of votes cast for each party ?

Solution : Each vote can be considered as a ball, which can have one of 15 possible colors. Hence we are in the situation of unordered choice with repitition allowed (unordered since we are only interested in the number of votes cast for each party, and not in the details of who actually votes for whom). In the notation of the previous proposition we have $n = 5,000,000$ and $k = 15$. Hence the number of possible outcomes is $\left(\begin{array}{c} 5,000,014 \\ 14 \end{array} \right)$.

Lecture 3 : Tuesday 12/9

We start with a reformulation of Proposition 9 which is sometimes easier to think about :

Proposition 10 *Let n, k be positive integers. The number of solutions to the equation*

$$x_1 + \cdots + x_k = n, \tag{13}$$

in which each x_i is a non-negative integer (i.e.: $x_i \geq 0$), is just $\left(\begin{array}{c} n+k-1 \\ k-1 \end{array} \right)$.

Proof : If, in choosing n balls in k different colors, we let x_i denote the number of balls chosen in color i, then we obtain an obvious 1-1 correspondence between the possible ways of choosing the balls and the solutions to (13). Thus Prop. 10 follows from Prop. 9.

This is a convenient point at which to introduce some terminology which may reappear later on, and which appears amongst the exercises handed out (see 10.2.16 and 10.2.17 for example) :

Definition 1 : A solution to (13) in which each x_i is strictly positive is called a *composition* of n with k *parts.*

Example : Take $n = k = 3$. By Prop. 10 there are $\left(\begin{array}{c} 3+3-1 \\ 3-1 \end{array} \right) = \left(\begin{array}{c} 5 \\ 2 \end{array} \right) = 10$ solutions in non-negative integers to $x_1 + x_2 + x_3 = 3$. You can write them all out, namely

$$\begin{array}{lll} 3+0+0 & 2+1+0 & 1+1+1 \\ 0+3+0 & 2+0+1 & \\ 0+3+0 & 1+2+0 & \\ & 1+0+2 & \\ & 0+2+1 & \\ & 0+1+2 & \end{array}$$

But only one of these is a composition of 3 into 3 parts, namely $1+1+1$. There are two compositions into 2 parts, namely $2+1$ and $1+2$, and one into a single part, namely 3 itself. Thus there are in all four compositions of the number 3.

A formula for the number of compositions of n into k parts can be deduced from Proposition 10 (how ?), and, summing over k, one gets a formula for the total number of compositions of n, which can be simplified to something very nice (see ex. 10.2.17). One should then try to interpret this formula combinatorially, i.e.: explain it directly, as in Example 4.

Balls and Bins

A common tongue-in-cheek description of the subject of combinatorics is that it is the science of throwing balls into bins. Jokes aside, there are many combinatorial problems (often of practical concern), which can be formulated in these terms. The basic question of interest is :

‘In how many ways can n balls be distributed among k bins ?’

By varying the conditions on how the balls may be distributed, and/or the nature of the balls and bins, one gets a range of possible questions, and it’s often not immediately obvious which questions will be easy, which will be really hard and which lie in the middle range which mathematicians like to describe as ‘interesting’.

We will be concerned with 4 variations of the basic question : we impose no conditions (for the moment) on the ways to distribute the balls, but consider the possibility that either the balls or the bins (or both) may be indistinguishable from one another : more concretely, they could be distinguished by having different colors, and are considered indistinguishable if they all have the same color. We have already developed the techniques to handle two of the resulting four variations, which we present now. The remaining two will be considered in subsequent lectures.

Variant 1 : *In how many ways can n mutually distinguishable balls be distributed among k mutually distinguishable bins ?*

Solution : Since all objects are mutually distinguishable, in order to have full information on the distribution of balls, one must know exactly in which bin each individual ball is placed. There are k choices for where to place each ball, hence, by MP, k^n choices for the entire distribution.

Answer : k^n.

VARIANT 2 : *In how many ways can n mutually indistinguishable balls be placed in k mutually distinguishable bins ?*

SOLUTION : If you think about it for a while, what is necessary to have full information in this case is simply knowledge of how many balls are placed in each individual bin. Which particular balls are placed in any bin is unimportant, so long as their number is known. Let x_i denote the number of balls in the i:th bin. Then we see that the possible ways to distribute the balls correspond naturally to the solutions of (13).

Answer : $\left(\begin{array}{c} n+k-1 \\ k-1 \end{array} \right)$.

INCLUSION-EXCLUSION (A.K.A. SIEVE) PRINCIPLE

Now for something rather different. The I-E principle is a very general (and occasionally useful) method for counting the elements in a finite union of finite sets when these sets overlap. If there was no overlap, then of course one would just count the elements in each set and add (the addition principle, basically). Otherwise, this will lead to an overcount, with elements that appear in two or more sets being overcounted.

Clearly, as the number of sets increases, so do the possibilities for overlapping and hence the complexity of handling this problem. Despite this, it turns out that there is a clear pattern in how the count should be performed in order to handle the overlaps.

To get a feeling for the problem, one can consider a small number of sets :

NOTATION : $|X|$ denotes the cardinality of (i.e.: number of elements in) the finite set X.

TWO SETS : Call them A and B. If we compute $|A| + |B|$ then elements in $A \cap B$ will have been counted twice. Hence, the I-E principle for two sets reads

$$|A \cup B| = |A| + |B| - |A \cap B|. \tag{14}$$

THREE SETS : Call them A, B and C. We could start by computing $|A| + |B| + |C|$. Anything present in exactly two of the three sets will have been counted twice. Hence we could continue by subtracting $|A \cap B| + |A \cap C| +$

$|B \cap C|$. But now consider an element present in all three sets : in the first step it is counted three times, in the second step it is removed three times. Hence it hasn't been counted at all. Thus we should go one step further and add back on $|A \cup B \cup C|$.

So the I-E principle for three sets reads

$$|A \cup B \cup C| = (|A| + |B| + |C|) - (|A \cap B| + |A \cap C| + |B \cap C|) + |A \cap B \cap C|. \tag{15}$$

Four Sets : Call them A, B, C and D. I'll leave it to yourself to work through the argument, and just state the result :

$$\begin{aligned} |A \cup B \cup C \cup D| &= |A| + |B| + |C| + |D| \\ &- (|A \cap B| + |A \cap C| + |A \cap D| + |B \cap C| + |B \cap D| + |C \cap D|) \\ &+ (|A \cap B \cap C| + |A \cap B \cap D| + |A \cap C \cap D| + |B \cap C \cap D|) \\ &- |A \cap B \cap C \cap D|. \end{aligned} \tag{16}$$

Hopefully the pattern is clear at this point, so we can state the general result :

Theorem 11 (I-E/Sieve Principle) *Let $A_1, ..., A_n$ be finite sets. Then*

$$\begin{aligned} \left| \bigcup_{i=1}^{n} A_i \right| &= \sum_{i=1}^{n} |A_i| - \sum_{i \neq j} |A_i \cap A_j| \\ &+ \sum_{i \neq j \neq k} |A_i \cap A_j \cap A_k| - \cdots + (-1)^{n-1} |A_1 \cap A_2 \cap \cdots \cap A_n|. \end{aligned} \tag{17}$$

Sketch Proof : One needs to show that every element of the union is counted exactly once on the right-hand side of (17), no matter how many of the sets A_i it appears in. Take any element of the union, call it x. Suppose x appears in k different sets, where $1 \leq k \leq n$. Then it is counted (or removed) once on the r.h.s. for each term of the form

$$A_{i_1} \cap \cdots \cap A_{i_l},$$

where $1 \leq l \leq k$ and the sets $A_{i_1}, ..., A_{i_l}$ are among the k sets containing x. The number of such terms, for a given l, is just $\left(\begin{array}{c} k \\ l \end{array} \right)$. Hence the total number of times x is counted on the r.h.s. is

$$\sum_{l=1}^{k} (-1)^l \left(\begin{array}{c} k \\ l \end{array} \right).$$

We want it to be counted exactly once, so this sum should equal 1. Noting that $C(k,0) = 1$, the resulting equation can be rewritten as

$$\sum_{l=0}^{k} (-1)^l \left(\begin{array}{c} k \\ l \end{array} \right) = 0. \tag{18}$$

See the exercise on page 8 for why (18) holds, for any $k \geq 1$.

We now proceed with some applications of the I-E principle.

Application I : The Euler phi-function

Definition 2 : Let n, m be two positive integers. The *greatest common divisor* of n and m is the largest positive integer which evenly divides both n and m. It is denoted $\text{GCD}(n,m)$. If $\text{GCD}(n,m) = 1$ then we say that n and m are *relatively prime (to one another).*

Example 6 : $\text{GCD}(2,3) = 1$ but $\text{GCD}(8,12) = 4$. Note that if p and q are any two distinct primes, then they are relatively prime.

Definition 3 : We define a function $\phi : \mathbf{N} \rightarrow \mathbf{N}$, called the *Euler phi-function*, as follows :

$$\phi(n) := \#\{x : x \in \mathbf{N}, 1 \leq x \leq n \text{ and } \text{GCD}(x,n) = 1\}. \tag{19}$$

In words, $\phi(n)$ is the number of integers, between 1 and n inclusive, which are relatively prime to n.

Example 7 : $\phi(6) = 2$. For among the number between 1 and 6, only 1 and 5 are relatively prime to 6. For the other numbers we have $\text{GCD}(2,6) = \text{GCD}(4,6) = 2$, $\text{GCD}(3,6) = 3$ and $\text{GCD}(6,6) = 6$.

The moral of the story which follows is that, with the help of the I-E principle, computation of $\phi(n)$ can be reduced to the problem of factorising n. Since, in general, no fast factorisation algorithms are known, this is in one respect not very satisfactory. However, it is precisely for this reason that the RSA cryptosystem is reasonably secure. Its security actually rests on the difficulty of computing the Euler function.

By the way, it is not known in general if there is some other, faster way to compute Euler-phi which avoids integer factorisation. It is generally believed, however, that there isn't, and in the special case applicable to RSA encryption (as we shall see later), it is quite easy to prove this.

We illustrate the method with an example, then state a general theorem. The proof of the theorem will be left to the reader.

EXAMPLE 8 : Compute $\phi(3000)$.

3000 is a pretty easy number to factorise, even by hand. We get that

$$3000 = 3 \cdot 1000 = 3 \cdot 10^3 = 3 \cdot (2 \cdot 5)^3 = 3 \cdot 2^3 \cdot 5^3.$$

From this and what's called the *Fundamental Theorem of Arithmetic* (something which everyone knows, but not many have seen a full proof of : we will discuss it more later), a number is relatively prime to 3000 if and only if it is not divisble by any of 2,3 and 5. This observation sets us up nicely for applying I-E. Define three sets :

$$\begin{aligned} A &:= \{x : 1 \leq x \leq 3000 \text{ and } x \text{ is a multiple of } 2\}, \\ B &:= \{x : 1 \leq x \leq 3000 \text{ and } x \text{ is a multiple of } 3\}, \\ C &:= \{x : 1 \leq x \leq 3000 \text{ and } x \text{ is a multiple of } 5\}. \end{aligned}$$

Then our observation above can be summarised as

$$\phi(3000) = 3000 - |A \cup B \cup C|.$$

So if we compute the size of the union, we're done. We use (15). Clearly,

$$|A| = \frac{3000}{2}, \quad |B| = \frac{3000}{3}, \quad |C| = \frac{3000}{5}.$$

What about $A \cap B$ for example ? Well, this set consists of numbers divisible by both 2 and 3. Another consequence[1] of the FTA is that this is precisely the same thing as saying that the numbers are divisible by $2 \cdot 3 = 6$. Hence

$$|A \cap B| = \frac{3000}{2 \cdot 3},$$

and similarly,

$$|A \cap C| = \frac{3000}{2 \cdot 5}, \quad |B \cap C| = \frac{3000}{3 \cdot 5}.$$

[1] The precise statement is as follows : if $\text{GCD}(n, m) = 1$ then a number is divisible by both n and m if and only if it is divisible by nm.

And applying the same reasoning to the intersection of all three sets, we get

$$|A \cap B \cap C| = \frac{3000}{2 \cdot 3 \cdot 5}.$$

Putting everything together and tidying up, we obtain

$$\phi(3000) = 3000 \times \left[1 - \left(\frac{1}{2} + \frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{2 \cdot 3} + \frac{1}{2 \cdot 5} + \frac{1}{3 \cdot 5}\right) - \frac{1}{2 \cdot 3 \cdot 5}\right].$$

We can tidy up further and write

$$\phi(3000) = 3000 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{5}\right) = 3000 \times \frac{1}{2} \times \frac{2}{3} \times \frac{4}{5} = 800.$$

Now for the general result :

Theorem 12

$$\phi(n) = n \times \prod_{p|n} \left(1 - \frac{1}{p}\right), \tag{20}$$

where the product is taken over the DISTINCT primes dividing n.

PROOF : Left as an exercise to the interested reader.

Lecture 4 : Friday 15/9

Application II of I-E : Derangements

Definition 4 : A *derangement* of the numbers $1, 2, ..., n$ is a permutation of them in which no number retains its place.

The number of derangements of n integers is denoted d_n. A priori, $d_n \leq n!$. Clearly, $d_1 = 0$ since you can't move just one number and $d_2 = 1$ since the only thing you can do with two numbers is switch them. One easily sees that the only derangements of 123 are the rotations 312 and 231, hence $d_3 = 2$.

Exercise : Write out all derangements of 1234.

Amongst the demonstration exercises, we proved the following two recursion formulas for d_n :

$$d_n = (n-1)(d_{n-1} + d_{n-2}), \ \ \forall\, n > 2, \tag{21}$$

$$d_n = nd_{n-1} + (-1)^n, \ \ \forall n > 1. \tag{22}$$

Eq. (22) in particular suggests strongly that d_n should be comparable in size to $n!$: the only difference with the recursion formula for the factorial function is the $(-1)^n$ term, plus we have a different initial condition in $d_1 = 0$ rather than 1. The following result is nevertheless satisfyingly precise :

Theorem 13

$$\lim_{n \to \infty} \frac{d_n}{n!} = \frac{1}{e}. \tag{23}$$

Remark : Intuitively, this result might be surprising. One might think that it should be very unlikely that a randomly chosen permutation of a large number of objects would have the property that not a single object retains its place. The theorem says that, on the contrary, the chances of this happening are about 36,8 percent.

Actually, there is a very simple probabilistic heuristic as to why a random permutation should have about a $1/e$ chance of being a derangement. We will mention this later, in the section on Discrete Probability. Here we give a rigorous proof of the theorem using Inclusion-Exclusion.

Proof of Theorem 13 : Fix n. Let X denote the set of all permutations of $1, 2, ..., n$, i.e.: of 1-1 functions $\pi : \{1, 2, ..., n\} \to \{1, 2, ..., n\}$. Hence $|X| = n!$. Define subsets $A_1, ..., A_n$ of X by

$$A_i := \{\pi \in X : \pi(i) = i\}, \qquad i = 1, ..., n.$$

Then, by definition,

$$d_n = n! - \left| \bigcup_{i=1}^{n} A_i \right|. \tag{24}$$

To compute the size of the union we use (17). First consider any A_i. The number i is left alone, and the remaining $n - 1$ numbers may be permuted freely amongst themselves. Hence, $|A_i| = (n - 1)!$. Thus the first sum in (17) becomes

$$\sum_{i=1}^{n} |A_i| = n \cdot (n - 1)! = n!$$

Next consider $A_i \cap A_j$ for any $i \neq j$. Both i and j are now left alone, and the remaining $n - 2$ numbers can be permuted freely amongst themselves, hence $|A_i \cap A_j| = (n - 2)!$. Thus the second sum in (17) becomes

$$\sum_{i \neq j} |A_i \cap A_j| = \left(\begin{array}{c} n \\ 2 \end{array} \right) \cdot (n - 2)! = \frac{n!}{2!(n - 2)!} \cdot (n - 2)! = \frac{n!}{2!}.$$

Let's do one more. For any $i \neq j \neq k$, $A_i \cap A_j \cap A_k$ consists of all permutations which leave i, j and k alone. Since the remaining $n - 3$ numbers can thus be permuted freely amongst themselves, we have $|A_i \cap A_j \cap A_k| = (n - 3)!$. So the third sum in (17) becomes

$$\sum_{i \neq j} |A_i \cap A_j \cap A_k| = \left(\begin{array}{c} n \\ 3 \end{array} \right) \cdot (n - 3)! = \frac{n!}{3!(n - 3)!} \cdot (n - 3)! = \frac{n!}{3!}.$$

Clearly, this all leads to the conclusion that

$$\left| \bigcup_{i=1}^{n} A_i \right| = n! \times \left(\sum_{k=1}^{n} \frac{(-1)^{k-1}}{k!} \right).$$

Substituting this into (24) and noting that $0! = 1$ we find that

$$\frac{d_n}{n!} = \sum_{k=0}^{n} \frac{(-1)^k}{k!}.$$

Hence

$$\lim_{n\to\infty} \frac{d_n}{n!} = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!},$$

provided the sum converges absolutely. But it does, and to e^{-1} as claimed, since the Taylor series for the exponential function

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

has infinite radius of convergence.

Discrete Probability Theory

When dealing with finite probability spaces (e.g: finitely many possible outcomes of an experiment), questions of probability can always be recast as combinatorial questions, i.e.: questions of counting. Let Ω be a finite probability space and A a subset of Ω. Then

$$P(A) = \frac{|A|}{|\Omega|}.$$

So to compute the probability of an event described by A one has to count the sizes of two finite sets, namely A and Ω, where the latter counts all possible outcomes, and the former counts those outcomes for which the desired event occurs.

Basic counting principles like the multiplication, addition and sieve principles can easily be recast in probabilistic terminology. Note, though, that these reformulations will still apply in any probability space whatsoever, not just a finite one.

Multiplication Principle *For any two events A and B in a probability space, we have that*

$$P(A \cap B) = P(A) \cdot P(B|A).$$

In particular, if A and B are independent events, then

$$P(A \cap B) = P(A) \cdot P(B). \tag{25}$$

Addition Principle *For any two mutually exclusive events A and B in a probability space, we have that*

$$P(A \cup B) = P(A) + P(B).$$

The sieve principle is then just a generalisation of the addition principle to overlapping events. So in the case of two events it would read

Sieve Principle *If A and B are any two events in a probability space then*

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Similarly, we obtain the probabilistic sieve principle for any finite number of events by simply replacing $|\cdot|$ by $P(\cdot)$ everywhere in (17).

EXAMPLE 9 : One can often get good insight into a combinatorial problem more quickly by thinking probabilistically. A good example of this is with derangements. It is very easy to see intuitively why Theorem 13 should hold. For consider a random permutation of the numbers $1, 2, ..., n$. Let A_i be the event that i gets moved. Clearly $P(A_i) = 1 - \frac{1}{n}$. Now if the events A_i were independent of one another, then by the multiplication principle (25) we'd have that

$$P(A_1 \cap \cdots \cap A_n) = \left(1 - \frac{1}{n}\right)^n.$$

But $A_1 \cap \cdots \cap A_n$ is just the event that a random permutation is a derangement and $(1 - \frac{1}{n})^n \to 1/e$ as $n \to \infty$, by the very definition of the exponential function.

The problem with this heuristic argument is, of course, that the events A_i are not independent. If they were, then we'd have $P(A_j|A_i) = 1 - \frac{1}{n}$. But a short calculation (left as an exercise to explain !) shows that

$$P(A_j|A_i) = \frac{n-2}{n-1} \times \frac{n-2}{n-1} + \frac{1}{n-1} \times 1 = \frac{n^2 - 3n + 3}{(n-1)^2}.$$

And another short calculation shows that this number is, in fact, slightly bigger than $1 - \frac{1}{n}$. In other words, if a random permutation is known to have moved one particular number, then it makes it slightly more likely that any other particular number will be moved (is this counter-intuitive ?). It is then very plausible indeed (but not yet proven !!) that the probability of a random permutation being a derangement should be at least $1/e$.

Balls and Bins II

We now return to the balls and bins problem and the remaining two variants of it. In each remaining case, the best we can do is get a fairly nice recursion formula for what it is we want to count, but only after adding an extra condition.

Variant 3 : *In how many ways can n mutually distinguishable balls be placed in k mutually indistinguishable bins ?*

Solution : No nice formula, or even recusrion formula, for the number of ways of doing this is known. Let's add another restriction, though. We set $S(n, k)$ to be the number of ways of distributing the balls in such a way that no bin is left empty. The numbers $S(n, k)$ are called *Stirling numbers of the second kind*[2]. We can obtain a nice recursion formula for them :

Theorem 14 *For any $n, k > 0$ we have that*

$$S(n, k) = k \cdot S(n-1, k) + S(n-1, k-1). \tag{26}$$

Proof : In this variant of the balls and bins problem, what is necessary to have full information is knowlegde of which balls are placed together : so it doesn't matter where a ball is placed as long as we know what other balls (if any) it is placed with. Focus attention on one of the balls and call it B. There are the following two alternative scenarios :

(i) B is placed alone in a bin

(ii) B has at least one binmate.

If (i) occurs then it doesn't matter which bin B is placed in. One bin is simply removed along with B, and it remains to place $n-1$ balls in $k-1$ bins, again under the restriction that no bin be left empty. By definition, there are $S(n-1, k-1)$ ways to carry out this placement.

[2] We ignore for the time being the obvious question as to what Stirling numbers of the first kind are.

If (ii) occurs, then it does matter where B is placed since it will have binmates. It is easier to think of first distributing the remaining balls, and then placing B. The remaining $n-1$ balls must be placed in k bins, and no bin can be left empty, as otherwise B would have to be placed in an empty bin and thus be alone. By definition, there are thus $S(n-1,k)$ ways to distribute the remaining balls. Now we place B, and there are k distinguishable choices for which bin to put it in. By MP, there are thus $k \cdot S(n-1,k)$ ways for the whole placement process, and the theorem is proved.

VARIANT 4 : *In how many ways can n indistinguishable balls be placed in k indistinguishable bins ?*

SOLUTION : As before there's no nice answer to this, and instead we impose a similar restriction. We set $p(n,k)$ to be the number of ways to distribute the balls so that no bin is left empty. The more common terminology is that $p(n,k)$ is the number of *partitions of the integer n into k parts.* This means that we write n as an unordered sum $n = x_1 + \cdots + x_k$ of k positive integers. We interpret x_i as the number of balls received by the i:th bin, and the indistinguishability of the bins is reflected in the fact that the sum is unordered, which means that we may interchange the summands without considering the partition of n as having beeen altered.

OBS! When writing partitions $n = x_1 + \cdots + x_k$, it is conventional to write the parts in decreasing order $x_1 \geq x_2 \geq \cdots \geq x_k$.

EXAMPLE 10 : $n = 7, k = 3$. We have $p(7,3) = 4$ since there are the following four partitions of 7 into 3 parts :

$$5+1+1 \quad 4+2+1 \quad 3+3+1 \quad 3+2+2.$$

As in the case of Stirling numbers, we can prove a nice recurrence for the partition numbers :

Theorem 15 *For any $n, k > 0$ we have*

$$p(n,k) = p(n-1,k-1) + p(n-k,k). \tag{27}$$

PROOF : Exercise.

Remark A function which has been studied extensively by mathematicians is the so-called *partition function* $p : \mathbf{N} \to \mathbf{N}$ given by

$$p(n) := \sum_k p(n,k).$$

In other words, $p(n)$ is the total number of partitions of n into any number of positive parts whatsoever.

Example 11 : $p(5) = 7$ as there are the following seven partitions of 5 :

$$5, \ \ 4+1, \ \ 3+2, \ \ 3+1+1, \ \ 2+2+1, \ \ 2+1+1+1, \ \ 1+1+1+1+1.$$

A really great theorem, proved after many, many years of trying, gives a very precise estimate of the growth rate of this function :

Theorem 16 (Hardy, Ramanujan, Rademacher 1937)

$$p(n) \sim \frac{e^{c_1\sqrt{n}}}{c_2 n}, \quad \textit{where } c_1 = \sqrt{\tfrac{2}{3}}\pi, \ c_2 = 4\sqrt{3}. \tag{28}$$

The proof of this theorem is way beyond the scope of this course.

Theorems 14 and 15 give example of recurrence relations which are very useful for computations, but from which it is very difficult to extract nice explicit formulas for, or even to estimate (without resort to computations) the growth rates of, the functions they describe. In the next lecture we will study a class of recurrence relations which are simple enough to be able to extract explicit formulas from them by a methodical procedure, but still of interest as they arise in fairly natural combinatorial settings.

Lecture 5 : Tuesday 19/9

We are going to study a class of 1-variable recurrence relations which are sufficiently simple to be able to extract explicit formulas from them for the number-sequences they describe, provided one can solve an algebraic equation. The general definition is as follows :

Definition 5 : Let $u_0, u_1, ...$ be a sequence of (a priori complex, though usually at worst rational) numbers. Let k be a positive integer. The sequence is said to satisfy a *linear recurrence relation of degree k with constant coefficients* if there are (complex) numbers $a_0, ..., a_k \neq 0$, and a function $f : \mathbf{N} \rightarrow \mathbf{C}$ such that, for all $n \geq k$,

$$a_k u_n + a_{k-1} u_{n-1} + \cdots + a_0 u_{n-k} = f(n). \qquad (29)$$

The recurrence (29) is said to be *homogeneous* if f is the zero function and *non-homogeneous* otherwise.

Observe that, since $a_k \neq 0$, if (29) is known to hold then u_n can be determined once $u_{n-1}, ..., u_{n-k}$ are all known. In particular, (29) determines the entire sequence of numbers (u_n) completely once one knows the values of $u_0, u_1, ..., u_{k-1}$. These starting values are referred to as the *initial conditions* attached to the recurrence relation.

We first concentrate on the homogeneous case. The basic theoretical result says two things :

I. One can write down a general formula for any sequence of numbers satisfying (29) with $f \equiv 0$, in terms of the roots of the degree k polynomial equation

$$a_k x^k + a_{k-1} x^{k-1} + \cdots + a_1 x + a_0 = 0. \qquad (30)$$

Eq. (30) is called the *characteristic equation* of the recurrence relation (29).

II. This formula will contain k free parameters. If one inserts the intial conditions into the formula then one obtains k linear equations for these k parameters. This linear system is guaranteed to have a unique solution.

Hence I and II imply that, so long as one has the starting values to hand and can solve an algebraic equation, one can write down an explicit formula for the entire sequence of numbers. Of course, there is a catch here, namely that algebraic equations like (30) are not easy to solve for any degree higher than 2. But solutions are easy to approximate very well, hence one can get 'approximate' explicit formulas. As such formulas typically involve exponentially growing terms, one is normally just interested in the rate of exponential growth (i.e.: the exponent), and this is obtained directly from the roots of the characteristic equation.

Rather than stating the general theorem, which is only likely to get one totally confused, partly because of all the notation, and partly because the exact formulation is a bit complicated to write down, let us illustrate the ideas in cases where we actually can solve the characteristic equation exactly, namely when the degree is at most 2. For the moment we remain in the homogeneous setting. Later we will also discuss non-homogeneou relations.

Degree one

Actually we didn't talk about this today, but will do so on Friday. It's natural to present the stuff here.

When $k = 1$, (29) can be rewritten in the form

$$u_n = c \cdot u_{n-1} \tag{31}$$

where c is some constant. In particular, $u_1 = cu_0$. Then $u_2 = cu_1 = c(cu_0) = c^2u_0$. Clearly, the general relation is that

$$u_n = c^n u_0, \quad \text{for all } n \geq 0. \tag{32}$$

This is an explicit formula for u_n involving one free parameter, namely u_0. If we know u_0 then we have an exact formula for u_n. Note that c is the unique root of the degree-1 characteristic equation, which is $x - c = 0$.

Example 11 : The example we've all seen before is getting interest on a bank investment. Suppose you deposit 100 crowns and get 5 percent compound interest per year. How much money have you got after 20 years ?

SOLUTION : Let u_n be the amount of dosh you've got after n years. Given is that $u_0 = 100$ and that

$$u_n = (1,05)u_{n-1} \quad \text{for any } n \geq 1.$$

So $c = 1,05$ here and the general formula reads $u_n = (1,05)^n u_0 = 100 \cdot (1,05)^n$. In particular, after 20 years we've got $u_{20} = 100 \cdot (1,05)^{20} \approx 265,3$ crowns.

DEGREE TWO

The degree one case was too easy to really see what's going on, but the essential features of the theory already become apparent in the degree two setting. Eq. (29) can now be rewritten as

$$u_n = au_{n-1} + bu_{n-2}, \quad \text{for all } n \geq 2. \tag{33}$$

The characteristic equation is

$$x^2 = ax + b. \tag{34}$$

This quadratic equation will have either one or two complex roots. Let α be any root of (34). The first essential observation is that if we set

$$u_n := \alpha^n \quad \text{for all } n \geq 0,$$

then (33) will be satisfied. Indeed the requirement is that

$$\alpha^n = a\alpha^{n-1} + b\alpha^{n-2}, \quad \text{for all } n \geq 2.$$

Dividing across by α^{n-2}, this reduces to $\alpha^2 = a\alpha + b$. But this just means that α should be a root of (34), which is precisely our assumption.

The second essential observation is that, if (u_n) and (v_n) are two sequences of numbers each satisfying (33), then the sequences (w_n) and (z_n) given by $w_n := Cu_n$ for some fixed constant C and $z_n := u_n + v_n$ also satisfy (33). This one readily checks. Speaking more formally, it means that any linear combination of solutions to (33) is also a solution. This is what is meant by the recurrence relation being 'linear'.

Let us now suppose first of all that the characteristic equation (34) has two distinct complex roots, which we call α and β. From our first observation, we see that both the sequences $u_n := \alpha^n$ and $u_n := \beta^n$ satisfy (33). From our second observation, we deduce that for any choice of complex numbers C_1 and C_2, the sequence

$$u_n := C_1\alpha^n + C_2\beta^n$$

will still satisfy (33). But this is the most general possible form of a solution. For C_1 and C_2 are free parameters. If we insert the inital conditions u_0 and u_1, then we get a system of two linear equations for these parameters, which can be written in matrix form as

$$\begin{pmatrix} 1 & 1 \\ \alpha & \beta \end{pmatrix} \begin{pmatrix} C_1 \\ C_2 \end{pmatrix} = \begin{pmatrix} u_0 \\ u_1 \end{pmatrix}.$$

The coefficient matrix has determinant $\alpha - \beta \neq 0$ hence there is always a unique solution for C_1 and C_2. Thus we get an explicit formula for the numbers u_n.

In the case when the characteristic equation has a single root α, one may check that not only $u_n := \alpha^n$ but also $u_n := n\alpha^n$ satisfies (33)[3]. Thus the most general possible solution in this case is

$$u_n := (C_1 + C_2 n)\alpha^n.$$

Insertion of the initial conditions determines C_1 and C_2 and thus gives a fully explicit formula for u_n.

Example 12 : We only did one example in class, the classical one of the so-called *Fibonacci numbers.*

First, returning to Example 11, observe that the compound interest model of investment growth can equally well be applied to other growth ($c > 1$) or decay ($c < 1$) processes, for example population growth ($c > 1$) or depreciation of an asset ($c < 1$). In this simplest of growth/decay models, which for obvious reasons is called the model of *natural growth/decay*, the quantity being studied grows or decays by a certain fixed factor c in each interval of time. This leads to exponential growth/decay at rate c. The

[3] Basically this corresponds to the fact that a polynomial $p(x)$ has a double root in $x = a$ if and only if its derivative $p'(x)$ also has a root in $x = a$.

continuous version of this model, which some of you may be more familiar with, is described by the differential equation

$$\frac{du}{dt} = c_1 u,$$

whose solution is $u_n = u_0 e^{c_1 t}$, giving exponential growth resp. decay at rate e^{c_1} when $c_1 > 0$ (resp. $c_1 < 0$).

Fibonacci was an Italian mathematician who lived in the 12th-13th centuries, and is one of the first important mathematicians of the modern West. His model for population growth, which also leads to exponential growth but is a bit more complicated than the basic natural growth model, is apparently based on his observations of populations of rabbits. The model contains a lot of simplifying assumptions, some of which are downright silly, but nevertheless the sequence of numbers which arises from it, the so-called *Fibonacci numbers*, have turned out to actually arise both in nature and in many combinatorial manifestations[4]. Let us now describe Fibonacci's model :

1. Rabbits come in male-female pairs.
2. Rabbits breed exactly one month after birth.
3. Rabbit pregnancy lasts excatly one month.
4. Each adult male rabbit mates with one adult female rabbit each month and they give birth to one male-female pair of twins.
5. Rabbits are immortal.

The last assumption is the most obviously ridiculous one (the others become less ridiculous if one thinks of them as statements of 'average' behaviour and demography in large rabbit populations, though admittedly the coincidence of the time frames in 2 and 3 seems somewhat arbitrary), but is the one which yields long-term exponential growth. Let's just accept the model and see what it gives.

So suppose you start off at time $t = 0$ with one pair of newborn rabbits. The basic question is : how many pairs of rabbits will you ave after n months ? Denote this number by F_n. So we're assuming $F_0 = 1$. Also $F_1 = 1$ since after one month we'll still have one pair, which are now fully grown. Then

[4] There is, in fact, an entire mainstream mathematical research journal dedicated to Fibonacci, called the *Fibonacci Quarterly.*

$F_2 = 2$ since that pair will now have given birth to a new pair. And so on. The important observation is that, for any $n > 1$,

$$F_n = F_{n-1} + F_{n-2}. \tag{35}$$

To see this, rewrite it as $F_n - F_{n-1} = F_{n-2}$. The left-hand side is the number of newborn pairs at the end of the n:th month. Each such pair was conceived by an adult pair at the end of the previous month. Thus the LHS equals the number of adult pairs at the end of the $(n-1)$:st month. But this in turn equals the total number of rabbit pairs at the end of the $(n-2)$:nd month, which is just by definition F_{n-2}. This proves (35).

Eq. (35) is a standard 2nd order linear recurrence with constant coefficients. The characteristic equation is $x^2 = x + 1$, which has roots $(1 \pm \sqrt{5})/2$. Hence, the most general possible solution to (35) is

$$F_n = C_1 \left(\frac{1+\sqrt{5}}{2} \right)^n + C_2 \left(\frac{1-\sqrt{5}}{2} \right)^n .$$

Inserting the initial conditions $F_0 = F_1 = 1$ and doing the linear algebra, we find that

$$C_1 = \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right), \qquad C_2 = -\frac{1}{\sqrt{5}} \left(\frac{1-\sqrt{5}}{2} \right),$$

and hence

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^{n+1} - \left(\frac{1-\sqrt{5}}{2} \right)^{n+1} \right]. \tag{36}$$

Some worthwhile remarks :

1. Despite the presence of the irrational number $\sqrt{5}$ in this formula, it must yield an integer value for every value of n. I leave it as an exercise for you to explain this purely algebraically.
2. Since $\left| \frac{1-\sqrt{5}}{2} \right| < 1$, the second exponential term in (36) will go to zero as $n \to \infty$. Thus, we have that

$$F_n \sim \frac{1}{\sqrt{5}} \left(\frac{1+\sqrt{5}}{2} \right)^{n+1} .$$

In particular, this implies that

$$\frac{F_n}{F_{n-1}} \sim \frac{1+\sqrt{5}}{2},$$

i.e.: the Fibonacci numbers grow at an exponential rate given by the so-called *golden ratio.* The ubiquity of the golden ratio in nature, and its aesthetic significance, is undoubtedly somehow tied up with the ubiquity of the Fibonacci numbers.
3. The sequence of Fibonacci numbers

$$1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 134, 223, ...$$

is one with which more or less every mathematician is familiar. If you want to get some idea of its ubiquity in combinatorics, go to

http://www.research.att.com/∼njas/sequences/

and type in the first few numbers of the sequence. For an application to which I am particularly partial, see

http://www.chlond.demon.co.uk/Queen.html

The problem there is to figure out from what starting positions on an infinitely large board the computer can always beat you. The Fibonacci numbers make an unexpected appearance somewhere !

Lecture 6 : Friday 29/9

We turn to inhomogeneous linear recurrences (29). The general result is the following :

Theorem 17 *Let $u^p = (u_n^p)$ be any solution of (29) and let $u^h = (u_n^h)$ be the general form of the solution to the homogenized equation obtained by setting $f \equiv 0$. Then the general form of the solution to (29) is*

$$u_n = u_n^h + u_n^p.$$

Proof : This follows from the linearity of the equation,which implies that if $\mathbf{u} = (u_n)$ and $\mathbf{v} = (v_n)$ are any two solutions of (29) then $\mathbf{u}$-$\mathbf{v} = (u_n - v_n)$ will satisfy the homogenized equation. The theorem follows.

We already know that solving homogeneous equations reduces to finding the roots of a polynomial. According to Theorem 17, in the inhomogeneous case, we have the extra task of finding any *particular* solution to the whole equation. For most functions $f(n)$ this will be a messy process. However, in two cases there will be such a particular solution of a standard form. Fortunately, these cases are also the most natural and interesting ones. They are

Case I : $f(n)$ is a polynomial.

Case II : $f(n) = a^n$, an exponential function.

In Case I, the generic choice for u_n^p is a polynomial of the same degree as f. The correct choice of coefficients is determined by insertion into (29). In Case II the generic choice is $u_n^p := C \cdot a^n$, with the same exponent a and a constant C which is determined by insertion into (29).

In either case, the only catch is that the generic choice for u_n^p may already be a solution of the homogenized equation, making (29) unsatisfiable if $f \not\equiv 0$. If so, then the second choice is $n\times$ (first choice). If this is still a solution of the homogenized equation (which can only happen if the characteristic equation has a repeated root), then the third choice is $n\times$ (second choice). And so on ... at some point you will be picking as your choice for u_n^p something which is not a solution of the homogenized equation, and then things will work out.

Finally, note that we can combine Cases I and II. If $f(n)$ is of the form

$p(n)a^n + q(n)$, where p, q are polynomials, then the generic choice for u^p is $u_n^p := P(n)a^n + Q(n)$ where P, Q are polynomials of the same degree as p, q respectively, with coefficients to be determined by insertion into (29).

EXAMPLE 13 (SEE 25.6.2 IN BIGGS) : Suppose you have a 4-letter alphabet, say $\{a, b, c, d\}$. Let q_n be the number of words of length n in this alphabet which contain an odd number of a:s. We want to find and solve a recurrence relation for q_n.

SOLUTION : Clearly, $q_1 = 1$ since the only such word of length 1 is a itself. And, for example, $q_2 = 6$ since exactly one of the two letters must be a, and then there are three choices for the other letter, thus giving the six words : ab, ac, ad, ba, ca, da.

Let $n > 1$. We divide the admissable words of length n into two types :

Type I : Those that begin with an a. Then amongst the remaining $n - 1$ letters there must NOT be an odd number of a:s. Since there are 4^{n-1} possible words of length $n - 1$ in all, exactly $4^{n-1} - q_{n-1}$ of these will not have an odd number of a:s. Thus there are this many words of Type I.

Type II : Those beginning with another letter. There are thus three choices for the first letter. Amonst the remaining $n - 1$ letters, there must still be an odd numebr of a:s. Thus there are q_{n-1} choices for the remaining letters. By MP, there are $3q_{n-1}$ words of Type II.

Adding up, we find that

$$q_n = (4^{n-1} - q_{n-1}) + 3q_{n-1} = 2q_{n-1} + 4^{n-1} \quad \forall\, n > 1. \tag{37}$$

This recurrence is of the form (29). The characteristic equation for the homogenized version is $x = 2$, so in this case

$$q_n^h = C_1 \cdot 2^n.$$

By the discussion above, the particular solution to (37) should have the form $q_n^p = C_2 \cdot 4^n$. Inserting into (37) we get

$$C_2 \cdot 4^n = 2C_2 \cdot 4^{n-1} + 4^{n-1},$$

and dividing across by 4^{n-1} yields $C_2 = 1/2$. Hence the general form of the solution to (37) is

$$q_n = q_n^h + q_n^p = C_1 \cdot 2^n + \frac{1}{2} \cdot 4^n.$$

Inserting the initial condition $q_1 = 1$ yields $C_1 = -1/2$, hence the explicit solution

$$q_n = \frac{1}{2} \left(4^n - 2^n\right).$$

Btw, note that this says that slightly less than half of all words of length n (for large n) contain an odd number of a:s. It would be interesting to have a simple intuitive 'combinatorial' explanation for this.

Examples of second degree inhomogeneous recurrences like (29) appear amongst the demonstration exercises.

Graph Theory

We change tack now for the second part of the course, and study a whole new topic, though still with a strong combinatorial flavour. Though graph theory has its origins in the work of famous mathematicians like Euler (mid 18th century) and Hamilton (mid 19th century), like other areas of discrete mathematics, it only began to be taken really seriously from WWII onwards, in parallel with the rapid development of computer technology. Many 'practical' problems can be modelled with graphs, though finding practically useful solutions usually involves finding some efficient algorithm for some procedure. With the development of computers, what counts as 'efficient' is constantly evolving, which in turn has spurred the constant search for new applications. Graph theory has also evolved as a sophisticated discipline within pure mathematics, and the range of research activity is now so varied that it is no longer really considered as just a subfield of combinatorics, though combinatorial methods still dominate.

We will have time for only a short introduction, emphasising some basic problems of both historical and practical significance.

Today we present the problem which is considered the historical origin of graph theory, and which gave rise to the first real 'theorem' in the field (Example 15 below). First, some formal definitions :

DEFINITION 6 : A *graph* G consists of the following data :

(i) a finite set V, whose elements are called the *vertices/nodes* of G.

(ii) a collection E of 2-element subsets of V. The members of E are called the *edges* of G.

A graph is easy to visualise. Think of the vertices as being points in space and the edges as joining pairs of points.

EXAMPLE 14 : Let $G = (V, E)$ be the following graph :

$$V = \{1,2,3,4,5,6\},$$
$$E = \{\{1,2\},\{1,5\},\{2,3\},\{3,4\},\{4,5\},\{4,6\}\}.$$

Pictorially, it's the graph on the right of the page at

http://en.wikipedia.org/wiki/Graph_theory

DEFINITION 7 : A *multigraph* is the same as a graph, except that in the edge-set E we allow repititions of the same 2-element subset of V. Visually, this just means that we allow a single pair of vertices to be joined by more than one edge. Note that this choice of terminology is not the universally accepted one. In some books, the word 'graph' includes what we have referred to as multigraphs. In such cases, what we have called graphs are referred to as *simple graphs*. It is crucial, therefore, when reading a text, to first be certain about what terminology they are using.

EXAMPLE 15 : This example is considered the historical origin of graph theory, and is commonly referred to as the *Königsberg bridge problem*. For pictures, see

http://mathworld.wolfram.com/KoenigsbergBridgeProblem.html

The drawing on the left of Fig. 98 there refers to a system of bridges and islands in the city of Königsberg, East Prussia, around the middle of the 18th century. There is a river flowing through the city center, in which are two islands. The islands are connected by a bridge. The larger island is also connected to the mainland by two bridges on each side, and the smaller one by a single bridge on each side. The problem is the following :

Is it possible to take a walk around the city and islands in such a way that one walks across every bridge exactly once ?

The essential features of the Königsberg bridge problem can be abstracted as a graph like that on the right of Fig. 98. Now the question becomes whether one can draw this graph without lifting one's pen from the paper. Let's state the question more formally :

DEFINITION 8 : A *path* in a (multi)graph is a sequence of edges such that the endpoint of any edge in the sequence coincides with the starting point of the next one. A path which begins and ends at the same vertex is called a *cycle/circuit.*

DEFINITION 9 : An *Euler path (resp. cycle)* in a (multi)graph is a path (resp. cycle) which includes every edge of the graph exactly once.

So, formally, the Königsberg bridge problem asks whether the graph on the right of Fig. 98 contains an Euler path or maybe even an Euler cycle. The answer is an emphatic NO. It is quite easy to give a simple characterisation of those (multi)graphs which contain such a path or cycle : Euler gets the credit for being the first person to formulate the criterion explicitly. Before stating the result, we need one further piece of terminology :

DEFINITION 10 : Let G be a (multi)graph and v a vertex in G. The *degree* of v, denoted $\deg(v)$, is the number of edges passing through v.

So the degrees of the vertices in the graph of Example 14 are, in numerical order, 2,3,2,2,3 and 1. The degrees in Example 15 are, in alphabetical order, 5,3,3 and 3.

Theorem 18 (Euler) *(i) A multigraph has an Euler cycle if and only if every vertex has even degree.*

(ii) A multigraph has an Euler path which is not a cycle if and only if exactly two of the vertices have odd degree. In this case, any Euler path must start at one of these two vertices and finish at the other one.

REMARK 1 : Note, in particular, that the theorem implies that a (multi)graph cannot possess both an Euler cycle and an Euler path which is not a cycle.

Remark 2 : Theorem 18 is nowadays also referred to as the *Postman Theorem.* A postman's route can be abstracted as a graph in which the vertices are street junctions and the edges streets. The postman would be delighted if the graph of his/her route had an Euler cycle as this would minimise the amount of walking he/she would have to do.

Proof of Theorem 18 (Part One) : It's very easy to prove the necessity of the conditions of the theorem. For if every edge is to appear in a path exactly once, then any time such a path enters a vertex which is neither a starting nor finishing point, then it must leave it again along a different edge. Thus every visit to a vertex, other than at the very start or very end of the path, will use up two of the edges through it.

It remains to show that a graph which satisfies the conditions of part (i), resp. (ii) of the theorem possesses an Euler cycle, resp. path. We will return to this issue next day. Note that one would ideally like a proof of these facts to be *constructive*, i.e.: it should not just show that an Euler cycle (resp. path) exists, but should indicate how to go about finding one. This is the *algorithmic* side of the Königsberg problem : to be wholly satisfied, one seeks an efficient algorithm for finding Euler cycles/paths in graphs which fulfill the requirements of Theorem 18. Note that the statement of Theorem 18 already gives an efficient algorithm for testing whether an Euler cycle/path exists : one just has to compute the degrees of all the vertices.

It turns out that a very simple-minded algorithm for finding Euler cycles/paths works.

Lecture 7 : Tuesday 27/9

A very simple kind of *Depth First Search* algorithm finds an Euler cycle in a graph in which every vertex has even degree. The same algorithm, with minor modifications, works in finding an Euler path in a graph with two vertices of odd degree. We describe the algorithm in the former case, then indicate the necessary modifications for the latter case.

Step 1 : Start anywhere and proceed along edges chosen at random until you reach a vertex where you're stuck and there's no way out. Now the point is that, since every vertex has even degree, the only place you can get stuck is back at the starting point. Thus you'll have traversed a cycle, which we'll call C_1.

Step 2 : When you get stuck, return along the path you came (the algorithm will thus require some memory) until you reach a vertex from which there remains an unused edge. If you find no such vertex, it means you've already covered all the edges and C_1 is an Euler cycle. Otherwise, starting at this vertex, which we'll call v, proceed at random along so far unused edges, until you get stuck again. Once again the point is that you can only get stuck back at v. Call the cycle traversed at this step C_2. Put C_1 and C_2 together as follows : first follow C_1 as far as v, then insert C_2 as a *detour*, before continuing along C_1 to the end. Update by calling this big cycle C_1 instead.

Step 3 : Repeat Step 2 as often as necessary until your cycle C_1 includes all the edges of G and hence is an Euler cycle.

In the case of a graph with two vertices of odd degree, the starting point in *Step 1* must be one of these two vertices. Then the first time you get stuck will have to be at the other such vertex. The remaining steps of the algorithm, involving the insertion of detours, are as above.

Hamilton Cycles

Definition 11 : A *Hamilton path (resp. cycle)* in a graph[5] is a path (resp.

[5] There is no loss of generality in discussing this problem only for graphs and not multigraphs, as any multigraph possessing a Hamilton cycle/path has a simple subgraph doing likewise. This need not be the case for Euler cycles/paths : give an example !

cycle) which visits every vertex excatly once (resp. visits every vertex exactly once before returning to its starting point).

At a first glance, the problem of deciding which graphs have Hamilton paths or cycles, and finding them when they exist, may look just as innocuous as the corresponding problem for Euler paths or cycles. Indeed, the former also has its origins in a single famous example, due to Hamilton (hence the name). In one of his lighter moments, when writing an article for a recreational mathematics journal (or something like that), he asked his readers to find a Hamilton cycle in the graph of a dodecahedron[6]

http://mathworld.wolfram.com/DodecahedralGraph.html

It's easy to find such a cycle by a bit of trial and error. However, for general graphs, one obviously wants something better than trial and error. Too bad !

Theorem 19 (1970s) *The problem of deciding whether an arbitrary graph has a Hamilton cycle or path is NP-complete.*

In a later lecture, we may or may not try to explain what this actually means[7]. In practice, though, what it means is that, unless some famous mathematical conjecture, the so-called $P \neq NP$ conjecture[8], turns out to be false, it is a hopeless task to find an efficient algorithmic procedure which can take an arbitrary graph as input, and output whether or not it contains a Hamilton cycle or path, never mind actually locate one. The precise statement is that no so-called *polynomial-time algorithm* for finding Hamilton paths/cycles exists if $P \neq NP$. This is a problem, because while it started off life as a toy problem, the Hamilton cycle problem is a special case of the important

Travelling Salesman Problem (TSP) *Given a connected, weighted graph,*

[6] The dodecahedron is one of the five *Platonic solids* or *regular polyhedra*, that is, polyhedra each of whose faces is a regular polygon with the same number of sides, and all of whose angles, both within any face and between any two faces, are equal. Later, we might show how the theory of *planar graphs* can be used to prove that there are only five such bodies (though the Greeks already could do it, obviously by other methods). For pictures of the Platonic solids, see http://en.wikipedia.org/wiki/Platonic_solid

[7] But you should take a CS course on 'Algorithms' or 'Complexity Theory' if you really want to understand.

[8] For more information, see http://www.claymath.org/millennium/

i.e.: a graph in which every edge has a non-negative real weight, find a path or cycle which visits every vertex at least once and whose total weight is as small as possible.

The basic Hamilton path/cycle problem is part of the special case where all the edges have the same weight. So noone knows of any general efficient procedure to solve TSP and there probably isn't any. A practically minded person could of course at this point say : well, ok, I can't usually find an optimal tour through a wieghted graph, but suppose I'm willing to live with something which I know is not far from optimal. Can I always find such a tour efficiently ? Here things get pretty interesting, it turns out. For general graphs, the answer is still no. For any fixed constant $c > 1$ there is no general polynomial-time algorithm (assuming $P \neq NP$) for finding a tour whose total weight is at most c times the minimum. However, if the graph lives in a real Euclidean space, and the weights represent distances between points (as in the case of a real-life travelling salesman, for example), then the situation is completely different : such polynomial-time algorithms do exist for ANY $c > 1$. There are also a bunch of results in between these two extremes, but apart from anything else, I don't know enough about the matter myself to go into them here.

Graph Colouring

Now for another innocuously playful problem which turns out to be very hard.

Definition 12 : A *(vertex) colouring* of a graph[9] is an assignment of a colour to each vertex in such a way that whenever two vertices are joined by an edge, they must get different colours. The *chromatic number* of a graph G, denoted $\chi(G)$, is the smallest number of colours needed to vertex-colour G.

Theorem 20 (1970s) *The problem of computing the chromatic number of a graph, and hence that of exhibiting an optimal vertex colouring of a graph, is NP-complete.*

As in the case of the Hamilton cycle problem, noone but a bunch of math weirdos and philosophers might care about this if it wasn't for the fact that

[9] Again, there's no loss of generality in only considering simple graphs here.

the coloring problem has a simple concretisation, namely to the problem of making *timetables*. Suppose, for example, someone (whom I'll call X to avoid any gender bias !) wanted to construct an examination timetable for all the students in TMA 055. X would first need a list of all the courses being taken by one or more students in the class. Then the basic requirements are (i) do not schedule two exams at the same time if there's at least one student taking both those courses (ii) minimise the total number of exam sessions, in order to minimise costs. The problem can be easily reformulated as a graph coloring problem : Let $G = (V, E)$ be the graph whose vertices are all the courses being taken by TMA 055 students, and whose edges join pairs of courses which have at least one student in common. Then the scheduler is interested in computing $\chi(G)$.

Definition 13 : The *complete graph on n vertices*, denoted K_n, is the graph on n vertices in which every pair are joined by a single edge. For some pictures, see http://mathworld.wolfram.com/CompleteGraph.html

For each $n \geq 3$, the *n-cycle*, or *cycle of length n*, denoted C_n, is the unique connected graph on n vertices in which every vertex has degree two. See http://mathworld.wolfram.com/CycleGraph.html

Note that K_3 and C_3 are the same graph, sometimes called the *triangle*.

Here are some simple observations about chromatic numbers, which are sometimes useful :

Proposition 21 *(i) If H is a subgraph of G, then $\chi(G) \geq \chi(H)$.*
(ii) $\chi(G) \geq 2$ if G contains any edges whatsoever.
(iii) $\chi(C_n) = 2$ if n is even and $\chi(C_n) = 3$ if n is odd.
(iv) $\chi(K_n) = n$.

Definition 14 : A graph with chromatic number 2 is called *bipartite*. The name is very suggestive, since if $\chi(G) = 2$ then it means that the vertices can be divided into two groups, such that there are no edges within either group, and all the edges cross from one group to the other.

Bipartite graphs arise in many situations and will be studied separately later on in the problem of *matchings*. For the moment, note that Prop. 21(iii) implies that a bipartite graph can have no cycles of odd length (this can also be deduced directly from the fact that all the edges cross between two disjoint groups of vertices). In fact, the converse is also true. This will be proven tomorrow.

Lecture 8 : 29/9

We conclude our discussion of graph colouring with a couple more easy results. The first is the promised converse to a corollary of Prop. 21(iii).

Proposition 22 *If the graph G has no cycles of odd length, then it is bipartite.*

PROOF : We describe an algorithm for an explicit 2-colouring of G. We call the colours 'red' and 'blue'.

Step 1 : Let $v \in V(G)$ be a randomly chosen vertex. Set $V_1 := \{v\}$. Colour v red.

Step 2 : Let V_2 be the set of neighbours of v. Colour all these vertices blue. Note that we can do this as if any two vertices a, b of V_2 were neighbours, then $v \rightarrow a \rightarrow b \rightarrow v$ would be a cycle of length 3 in G.

Step 3 : Let V_3 denote the set of all neighbours of all vertices in V_2, other than v. Colour all the vertices of V_3 red. Again this is ok. For if two vertices $a, b \in V_3$ were neighbours then either

(i) both are neighbours of the same vertex w in V_2, in which case $w \rightarrow a \rightarrow b \rightarrow w$ would be a cycle of length 3 in G,

(ii) a and b are neighbours of distinct vertices w_1 and w_2 respectively in V_2. In this case, $v \rightarrow w_1 \rightarrow a \rightarrow b \rightarrow w_2 \rightarrow v$ would be a cycle of length 5 in G.

Step 4 : I think it is now obvious how the algorithm will proceed and hence will not bother to describe any further steps.

For the second result, note that Proposition 21 essentially gives a variety of ways of obtaining lower bounds on the chromatic number of a graph. A simple upper bound is given by

Proposition 23 *For any graph G,*

$$\chi(G) \leq 1 + \Delta(G), \tag{38}$$

where

$$\Delta(G) := \max_{v \in V(G)} \{deg(v)\}.$$

Proof : Apply the following *greedy algorithm* to colouring the vertics of G :

(i) order the vertices arbitrarily as $v_1, ..., v_n$.
(ii) order available colours (assume there's an unlimited number of them) arbitrarily as $C_1, C_2, ...$
(iii) colour the vertices in order. At each vertex use the first colour which has not already been used to colour one of its neighbours.

It is clear that rule (iii) gurantees that no more than $1 + \Delta(G)$ colours will be used in this procedure, which completes the proof of the proposition.

The problem with the greedy graph-colouring algorithm is, as we have shown in the demonstrations, that different numbers of colours may be used depending on how the vertices are ordered. It is quite easy to show (exercise !) that there is always SOME ordering of the vertices for which the algorithm uses exactly $\chi(G)$ colours. But a priori there are $n!$ different orderings of n vertices, so this isn't going to lead to any generally efficient graph-colouring procedure.

Remark The *girth* of a graph G is the smallest length of a cycle in G, or $+\infty$ if G contains no cycles. It seems intuitively very reasonable that graphs of high girth should have low chromatic number, because one avoids dense clusters of vertices which might require many colours. In fact, for several decades in the middle of the last century a well-known conjecture asserted that there should be a universal upper bound on the chromatic number of graphs with sufficiently large girth. This conjecture was disproven by Erdős[10] in 1959. Using probabilistic methods, he proved the existence of graphs with arbitrarily large girth and chromatic number. In fact, he showed that in some sense it is quite normal for graphs of large girth to also have large chromatic number, even if it's atrociously difficult to give explicit examples. For a discussion of his result, see for example my lecture notes on Probabilistic Combinatorics, which I've posted on the course homepage.

[10] This Hungarian was the 20th century's most prolific mathematician, with over 1400 published research papers.

On the other hand, there is one famous large class of graphs all of which have low chromatic number : see Theorem 26 below.

Trees

Definition 15 : A *tree* is a connected graph without cycles[11].

Depending on which book you look in, you might see at least two alternative definitions of a tree. Of course, all definitions must be equivalent :

Proposition 24 *The following are equivalent for a graph G :*

(i) G is a tree, as defined above.

(ii) G is connected and, given any two vertices in G, there exists a unique path between them.

(iii) G is connected and the number of vertices in G is one more than the number of edges.

Proof : (i) $\Rightarrow$ (ii) since if there were two distinct paths between the same pair of vertices, then the graph would have to contain a cycle. Running the argument backwards, (ii) $\Rightarrow$ (i).

The intuitively easiest way to see that (i) $\Leftrightarrow$ (iii) is to think of growing the tree one edge at a time. We start off with a single vertex and no edge. As there are to be no cycles, for every edge added, we must also add a new vertex.

We will be discussing two applications of trees :

(A) So-called *Decision Trees* or *Branching Processes*

(B) A variety of problems involving the search for a *spanning tree* in a weighted (di)graph which is optimal in some specified sense.

First, though, motivated by Prop.24(iii), I wish to make a detour into the world of

Planar Graphs

Definition 16 : A graph G is said to be *planar* if it can be drawn in a plane

[11] Note that, in particular, a tree can have no multiple edges.

without any two edges crossing. Any such drawing of a graph is called, by a small abuse of terminology, a *plane graph*. Thus a plane graph is not a graph, but rather a particular drawing of a planar graph. If you know what I mean ....

EXAMPLE 16 : The drawing of K_4 at Mathworld (see Definition 13) is not plane, but can easily be made so by moving one of the diagonal edges to the outside. Thus K_4 is planar. On the other hand, one can check that K_5 is not. For another example of a non-planar graph we need

DEFINITION 17 : Let m, n be two positive integers. The *$m \times n$ complete bipartite graph* is the bipartite graph with m red and n blue vertices in which every red vertex is joined to every blue vertex. It is denoted $K_{m,n}$. Observe that $K_{m,n}$ has mn edges.

EXAMPLE 16 (CTD.) $K_{3,3}$ can be checked to be not planar.

The first big result on planar graphs is

Theorem 25 (Kuratowski's Theorem, 1930s) *The graph G is not planar if and only if it contains either K_5 or $K_{3,3}$ as a minor.*

Here we are using a previously undefined word, namely *graph minor*. The precise definition of this term is a bit technical, but basically a graph H is a minor of a graph G if H is a subgraph of G, or can be obtained from such a subgraph by *contracting* one or more edges. Contracting an edge means identifying the two vertices at its ends, and letting all edges protruding from either vertex be common to the newly identified vertex.

The second major result about planar graphs is

Theorem 26 (Four Colour Theorem, 1976) *If G is a planar graph, then $\chi(G) \leq 4$.*

Theorem 26 is generally regarded as the first major mathematical result proven with significant help from computers. The authors of the proof had an argument which reduced it to verifying the four-colourability of a large, but finite number of specific graphs. They got a computer to do this. Note that since there are no generally efficient graph colouring algorithms, one can imagine that this was a considerable task for the computer, especially back in the Dark Ages of 1976 !!

Prop. 24(iii) above is a special case of a very old result about plane graphs. Any plane graph divides the plane into a finite number of disjoint regions, namely the infinite region surrounding the outside of the graph, plus a number of enclosed regions.

Theorem 27 (Euler) *Let G be a plane graph. Let V, E, R denote respectively the number of vertices of G, the number of edges of G and the number of regions into which G divides the plane. Then*

$$V - E + R = 2. \tag{39}$$

NOTE : (39) is called *Euler's formula for plane graphs.* In more modern terminology, the alternating sum on the left-hand side is called an *Euler characteristic* and one says that the Euler characteristic of the plane is 2[12].

PROOF OF THEOREM 27 : Note that Prop. 24(iii) is a special case, since for a tree $R = 1$ (no cycles means no enclosed regions), and the proposition says that $V - E = 1$. The proof of the general theorem is the same as in this special case, namely we think of growing the plane graph one edge at a time. At the outset we have $V = R = 1$, $E = 0$, so (39) holds. Suppose k edges have been grown and that (39) holds. When the next edge is added, there are two possibilities :

(i) a new vertex is also grown. In this case, both V and E increase by one, but R is unchanged. Thus (39) still holds.

(ii) no new vertex is added, i.e.: the new edge goes between two existing vertices. It will thus create a new enclosed region, but since it cannot cross any other existing edges, it will create precisely one new region. Thus E and R both increase by one in this case, with V unchanged, so (39) still holds. This completes the proof.

A very nice application of Euler's formula is to prove that there are no

[12] The general modern result states that for any graph drawn on a surface in such a way that it divides it into simply connected regions, the Euler characteristic will always be the same number. Which number depends on the surface, more precisely on how far the surface itself deviates from being simply connected. Remember that a surface is said to be *simply connected* if any closed curve drawn in it can be contracted to a point without leaving the surface. So, for example, an annulus or doughnut is not simply connected.

more Platonic solids than the known five. Let S denote a Platonic solid and let G_S be the canonical representation of it by a plane graph : intuitively, this is obtained by punching a hole in one face of the solid, and drawing that face out so as to flatten the solid into the plane, in which case the destroyed face becomes the infinite region external to the graph. Let V, E, R have their usual meaning for this graph. We make two observations :

(i) First, the regularity of the solid means all vertices will have the same degree, say d. For any graph G,we have the relation

$$\sum_{v \in V(G)} \deg(v) = 2 \cdot |E(G)|, \tag{40}$$

since every edge is counted twice in the sum on the left. For G_S this implies that

$$dV = 2E \Rightarrow V = \frac{2}{d}E. \tag{41}$$

(ii) Suppose the faces of S are n-gons. Consider the following sum, which we denote by Σ : for each region of G_S, including the infinite region, count the surrounding edges, and then add. By assumption, $\Sigma = nR$. On the other hand, in this sum each edge of G_S will be counted twice, as it is shared between exactly two regions (because of planeness and the absence of edges 'sticking out' into the infinite region). Thus $\Sigma = 2E$. It follows that

$$R = \frac{2}{n}E. \tag{42}$$

Now (39), (41) and (42) imply that

$$2 = E \cdot \left(\frac{2}{d} - 1 + \frac{2}{n}\right). \tag{43}$$

In particular,

$$\frac{2}{d} - 1 + \frac{2}{n} > 0. \tag{44}$$

But $d \geq 3$ a priori, since you won't ba able to get a closed solid otherwise. Thus for $n \geq 6$ there are no possible solutions to (44). If $n = 5$ then the only solution is $d = 3$, and substituting back into (43), (42) and (41) yields $V = 20, E = 30, R = 12$. This is the *dodecahedron*. Similarly, if $n = 4$ then $d = 3$ is still the only possibility for (44), and substituting backwards yields $V = 8, E = 12, R = 6$. This is the *cube*.

If $n = 3$ then (44) is satisfied for $d = 3, 4$ or 5. Substituting backwards yields respectively

$V = 4, E = 6, R = 4$. This is the *tetrahedron*.
$V = 6, E = 12, R = 8$. This is the *octahedron*.
$V = 12, E = 30, R = 20$. This is the *icosahedron*.

Branching Processes

Definition 18 : A *rooted tree* is a tree in which some specific vertex has been designated as the root of the tree. Thus one imagines the tree as growing outwards from this root.

Definition 19 : A *leaf* in a tree is a vertex of degree one.

Visually, it is pretty clear that any tree must have at least two leaves. Note that this also follows rigorously from Prop. 24(iii) and eq.(40), since for a tree the latter reads

$$\sum_{v \in V(G)} \deg(v) = 2E = 2(V - 1) = 2V - 2,$$

and since the sum contains V terms, each a positive integer, at least two of these terms must equal one.

By Prop. 24(ii), in any tree it makes sense to talk about the *distance* between two vertices, as this is unambiguously defined as the length of the unique path between them[13]. In a rooted tree, let d_v denote the distance of a vertex from the root. Thus $d_v = 0$ if v is itself the root and $d_v > 0$ otherwise.

[13] For general graphs, one usually defines the distance between two vertices as the smallest length of a path between them. If there is no such path, i.e.: if the two vertices belong to different components of a disconnected graph, then one sets the distance between them to be $+\infty$. One writes $d(v, w)$ for the distance between vertices v and w. Note that the function $d : V(G) \times V(G) \to \mathbf{Z}_{\geq 0}$ is a *metric*, i.e.: it satisfies the following properties :
(i) $d(v, w) = 0$ if and only if $v = w$.
(ii) $d(v, w) = d(w, v)$.
(iii) For any three vertices v, w, x we have

$$d(v, x) \leq d(v, w) + d(w, x). \tag{45}$$

Relation (45) is called the *triangle inequality*.

Proposition 28 *Let G be a rooted tree and let v be any non-root vertex. Then*

(i) there is a single vertex w among the neighbours of v such that $d_w = d_v - 1$.

(ii) all other neighbours w' of v will satisfy $d_{w'} = d_v + 1$.

Proof : Exercise.

The vertex w in the above proposition is called the *parent* of the vertex v and the vertices w' are called the *children* of v. Note that v has no children if and only if it is a leaf.

The terminology is obviously suggestive of the branching process described by the usual *family tree*. Such branching processes are obviously central to population studies in biology and are evidently modelled by rooted trees. In the theory of algorithms, the sequence of steps in an algorithm can often be modelled by a rooted tree : the root represents the first step and, in general, the children of any node describe the possible outcomes of a particular step, which determine what the next step will be. In this context one usually speaks of *decision trees*.

I don't intend to spend much time studying applications of rooted trees as this runs into whole new fields and it is more useful to undertake the study within the context of a course in the particular field. Plus we don't have time. I will however, present one particularly cute toy example of an application of decision trees. Though we started with this today, I will present the material along with the notes for the next lecture.

Lecture 9 : Tuesday 3/10

DEFINITION 20 : The *height* of a rooted tree is the greatest distance of a vertex from the root.

DEFINITION 21 : Let n be a positive integer. An *n-ary rooted tree* is a rooted tree in which every vertex which is not a leaf has exactly n children.

Proposition 29 *Let n, l be positive integers. If G is an n-ary rooted tree with l leaves, then the height of G is at least $\lceil \log_n l \rceil$.*

EXAMPLE 17 (DEFECTIVE COIN PROBLEM) : Suppose we have N coins, all of which look identical, but one of which is known to be defective, either being slightly lighter or heavier than the others. We want to find the defective coin by a sequence of weighings on a balance scale. Any procedure for doing this can be represented as a ternary rooted tree. The non-leaf nodes represent the weighings performed, and the three children of each such node represent the three possible outcomes of this weighing : left side heavy, right side heavy, both sides equal. Each leaf node represents a possible outcome of the procedure, though there may be extra leaves corresponding to impossible chains of events. Since there are $2N$ possible outcomes - which coin is defective and whether it is light or heavy - the tree must have at least $2N$ leaves. By Prop. 29, the height of the tree is at least $\lceil \log_3 2N \rceil$, and this is thus a lower bound for the number of weighings required in a general procedure for revealing the defective coin.

For example, if $N = 12$ then $\lceil \log_3 24 \rceil = 3$. One of the homework exercises was to construct a decision tree for 12 coins. Note that the lower bound above is not always achieved : for example, I think (not 100 procent sure) there is no procedure for 13 coins which always requires at most three weighings. You can check for yourself if you're interested : this is a classical problem which I believe has been solved in full and also generalised. There are many references available online, but I am not familiar with them.

SOME OPTIMISATION PROBLEMS IN WEIGHTED GRAPHS

We will discuss the following three optimisation problems in weighted graphs :

(i) minimal spanning tree (MST) problem

(ii) shortest path problem
(iii) maximal flow problem in networks.

Minimal spanning trees

Definition 22 : A *spanning tree* in a connected graph is a subgraph which is a tree and which includes all the vertices of the graph.

A spanning tree of minimal total weight in a connected, weighted graph is called a *minimal spanning tree.*

There are two well-known algorithms for finding MST:s in weighted graphs :

Kruskal's Algorithm *Start anywhere. At each step choose the cheapest edge among those which go between a vertex already reached and one not yet reached. Choose arbitrarily whenever a choice is available. Continue until all vertices reached.*

Prim's Algorithm *At each step choose the cheapest remaining edge among those whose inclusion does not create a cycle. Choose arbitrarily whenever a choice is available. Continue until the number of edges chosen is one less than the number of vertices in the graph.*

Clearly, the algorithms are quite different : in particular, the first one grows a tree step-by-step, whereas the second grows a forest which becomes a single tree by the end. This must be proven of course, along with the fact that both algorithms are guaranteed to produce a MST. We omit these proofs however. Note that it can be shown that both algorithms have approximately the same time complexity. It is probably the case that they are essentially optimal in this regard, though I do not know if such facts have been proved.

Example 18 : The algorithms are best illustrated by doing examples. One was done in class : details omitted here.

Shortest path

It is pretty obvious what is meant by a shortest path between two vertices in a weighted graph. The standard algorithm for finding one is described below : note that it works even for *digraphs*, i.e.: graphs in which each edge is assigned a direction (hence can be thought of as an ORDERED pair of vertices), and movement along the edge is only allowed in the assigned direction.

Dijkstra's Algorithm *Suppose v and w are two nodes in a (di)graph G and we want to find a shortest path from v to w. Starting at v, at each step choose amongst those edges going from a node already reached to one not yet reached, that which minimises the total distance of the target node from v. Choose arbitrarily whenever a choice is available. Update the distance of the target node from v with this value. Continue until w becomes the target. Since the sequence of edges chosen form a tree, there is a unique path in this tree from v to w, which can be found by a simple depth-first search.*

Example 19 : Again we omit a proof that Dijkstra's algorithm works and concentrate on illustrating the procedure with examples. The details of the example presented in class are omitted.

Maximal flow

Definition 23 : A *network* is a connected digraph without directed cycles.

It is easy to see that in a network, there must exist at least one node for which all the vertices through it are directed outwards, and at least one node for which all the vertices through it are directed inwards. For otherwise, we could simply wander around the network and would eventually have to come back to a node already visited.

A node of the former type is called a *source* and one of the latter type is called a *sink*. For simplicity, we'll assume all our networks contain exactly one source and one sink. The Ford-Fulkerson algorithm, described below, can be generalised to networks with multiple sources and/or sinks, by identifying all the sources (resp. sinks) into a single so-called *supersource* (resp. *supersink*).

Definition 24 : In a network, the weight of an edge is usually referred to as its *capacity*. The capacity of an edge e is usually denoted $c(e)$.

Definition 25 : A *flow* in a network G is a function $f : E(G) \rightarrow [0, \infty)$ from the edge set of G to the non-negative reals satisfying the following two properties :

(i) (Admissability) For every edge e, $f(e) \leq c(e)$. In words, the flow along any edge cannot exceed its capacity.

(ii) (Mass conservation) For any vertex v which is neither the source nor the sink, the sum of the flows along the incoming edges must equal the sum of the flows along the outgoing edges.

Definition 26 : The *strength* of a flow f, denoted $|f|$, is the total flow out from the source, which, by mass conservation, equals the total flow into the sink. A flow whose strength is as large as possible is called a *maximal flow*.

Definition 27 : A *cut* in a network, with source s and sink t, is a partition of the vertices of the network into disjoint sets S and T such that $s \in S$ and $t \in T$. Cuts are denoted (S, T).

Definition 28 : Let (S, T) be a cut in a network. The *capacity of the cut*, denoted $c(S, T)$, is the sum of the capacities of the edges $e \in E(G)$, such that e is directed from a vertex in S to a vertex in T. A cut of smallest possible capacity is called a *minimal cut*.

Theorem 30 (Ford-Fulkerson, 1950s) *In any network, the maximum possible strength of a flow equals the minimum possible capacity of a cut.*

Remark : Philosophically, this is an example of a so-called *Max-Min theorem*, which says that the maximum of one particular quantity equals the minimum of another. Such theorems are ubiquitous in many areas of optimisation, for example in linear programming and game theory. In fact, the Ford-Fulkerson theorem is a special case of a result in linear programming.

Proof of Theorem 30 : One half of the theorem is trivial : mass conservation implies immediately that the strength of any flow cannot exceed the capacity of any cut. For our purposes, what is interesting about the rest of the proof of the theorem is that it is constructive, i.e.: it gives an efficient algorithm for constructing an explicit flow and cut such that the strength of the former equals the capacity of the latter. The essential idea in this construction is that of an *augmenting path* :

Definition 29 : Let f be a flow in a network. An *f-augmenting path* is a path between the source and the sink in the underlying undirected graph

such that, for every edge e on this path :
(i) if in the network, e is directed in the same direction as the path, then $f(e) < c(e)$,
(ii) if in the network, e is directed in the opposite direction to the path, then $f(e) > 0$.

An edge of the first type is said to be directed *forwards* and one of the second type directed *backwards*. All of this is relative to a given path, of course : the same edge could be directed forwards and backwards along different paths. For a forward-directed edge e along an augmenting path, set $\epsilon_e := c(e) - f(e)$. For a backwards-directed edge, set $\epsilon_e := f(e)$. Then set ϵ to be the minimum of the ϵ_e. Thus, for any f-augmenting path, $\epsilon > 0$.

Now the point is the following : Given a flow f, suppose we can find an f-augmenting path, with corresponding $\epsilon > 0$. Then f can be replaced by a stronger flow, by

(a) increasing the flow along every forward-directed edge of the augmenting path by ϵ

(b) removing entirely the flow along every backward-directed edge of the augmenting path

(c) leaving the flow unchanged along all remaining edges of the network.

The *Ford-Fulkerson algorithm* for finding a maximal flow in a network proceeds as follows :

Start with an entirely empty flow. At each step, search for an augmenting path by a usual search procedure (as in Kruskal's algorithm, for example). If you find such a path, then increase the flow as described above and repeat the procedure. The algorithm stops when no augmenting path can be found. When this happens, the vertices of the network will be partitioned into two subsets S and T by the failed search for an augmenting path : S will consist of all those vertices which could be reached by the search procedure via an augmenting path, and T will consist of the remaining vertices. Obviously, (S, T) is thus a cut.

What needs to be proven is that the strength of the final flow f equals the capacity of this cut (S, T). By mass conservation, $|f|$ equals the total flow along edges crossing from S to T minus that along edges crossing from T back into S. If there was any positive flow along an edge of the latter type, then this could be considered as a backward-directed edge along a path extending into T, and we'd have an augmenting path extending into T - contradiction. Similarly, if along any edge of the former type, the flow was less than the capacity, then we could consider this as a forward-directed edge along an augmenting path extending into T. Thus neither situation arises which, by the various definitions, means that $|f| = c(S,T)$. This completes the proof of the theorem.

EXAMPLE 20 : An example will be done in the exercise session on Thursday.

Lecture 10 : Friday 6/10

The subject of today's lecture is *matchings* in graphs, in particular bipartite graphs. Though it may not be immediately obvious, we will show that there is a close connection between this material and the network flows discussed previously.

DEFINITION 30 : A *matching* in a graph G is a subset M of $E(G)$ such that no two edges in M have a vertex in common. If $\{v, w\} \in M$, then v and w are said to be *matched* by M.

The *size* of a matching M is the number of edges in it, and is denoted $|M|$. A matching is said to be *perfect* if every vertex of G appears in some edge of the matching. Obviously, in this case, G has an even number of vertices and $|M| = \frac{1}{2}|V(G)|$.

DEFINITION 31 : Let $B = (X, Y, E)$ be a bipartite graph. A matching M is said to be *perfect for X (resp. Y)* if every vertex of X (resp. Y) is matched. Clearly, a necessary condition for there to exist such a matching is that $|X| \leq |Y|$ (resp. $|Y| \leq |X|$). Thus a necessary condition for the existence of a perfect matching for the whole graph is that $|X| = |Y|$.

The general problem we wish to study is :

'*Given a graph G, find a matching of maximum possible size in G*'.

The matching problem is most natural in the bipartite setting.

EXAMPLE 21 : Let $B = (X, Y, E)$ where X is a set of men, Y a set of women and $\{x, y\} \in E(B)$ if and only if man x is a compatible marriage partner for woman y. The *marriage problem* (yes, this is the official name !) asks to find a maximum size matching in such a bipartite graph.

For a less contraversial example, take $B = (X, Y, E)$ where X is a set of job seekers, Y a set of available jobs, and $\{x, y\} \in E(B)$ if and only if person x is qualified to do job y. The government would certainly appreciate having an efficient procedure for finding a maximum size matching in such a graph.

It turns out that there is a general, efficient procedure for finding maximum size matchings in graphs. We shall proceed as follows :

(i) we state and prove a necessary and sufficient criterion (Theorem 31) for the existence of a perfect matching for one side of a bipartite graph. The proof is constructive, i.e.: it describes an efficient algorithm for locating a perfect matching in a bipartite graph which satisfies the criterion.

(ii) we recast the above algorithm in terms of the Ford-Fulkerson algorithm for network flows.

(iii) we show how the algorithm can be extended to work in any graph, not just bipartite graphs, even if the interpretation in terms of network flows no longer makes sense.

Before stating the theorem, we need some more notation :

Notation : Let $B = (X, Y, E)$ be a bipartite graph and let A be a subset of X. We define

$$\Gamma(A) := \{y \in Y : \{a, y\} \in E(B) \text{ for some } a \in A\}.$$

In words, $\Gamma(A)$ is the set of vertices in Y which are neighbours to one or more vertices in A.

Theorem 31 (Hall's Marriage Theorem) *Let $B = (X, Y, E)$ be a bipartite graph. There exists a perfect matching for X if and only if $|\Gamma(A)| \geq |A|$ for all subsets A of X.*

Proof : Hall's condition is obviously necessary since a vertex in X can only be matched to one of its neighbours, so if every vertex is to be matched, every subset of vertices in X must have at least as many neighbours in all as there are vertices in the subset.

Now suppose Hall's condition is satisfied. Let M be any matching which is not perfect for X. We show how to construct a matching M^* such that $|M^*| = |M| + 1$.

Let x_0 be any vertex in X left unmatched by M. By Hall's criterion, $|\Gamma(\{x_0\})| \geq 1$, thus x_0 has at least one neighbour in Y. Pick any such neighbour y_0. If y_0 is also left unmatched by M, then just add the edge $\{x_0, y_0\}$ to M to obtain M^*. Otherwise, we may assume y_0 is already matched, to a vertex which we call x_1. Applying Hall's criterion again, we have that $|\Gamma(\{x_0, x_1\})| \geq 2$. Thus there is at least one more vertex $y_1 \in Y$ which is a neighbour to either x_0 or x_1. Two cases now arise :

Case I : y_1 is left unmatched by M. If y_1 is a neighbour of x_0 then add the edge $\{x_0, y_1\}$ to M to obtain M^*. If y_1 is a neighbour of x_1 then consider the following path in B from y_1 back to x_0 :

$$y_1 \to x_1 \to y_0 \to x_0.$$

The first and last edges of the path lie outside M, whereas the middle edge is in M. We now convert M into M^* by removing the edge $\{x_1, y_0\}$ from the matching and adding on the edges $\{y_1, x_1\}$ and $\{y_0, x_0\}$.

Case II : y_1 is already matched. The previous assumptions imply that it is matched to a new vertex x_2. The argument now proceeds as before. Hall's criterion applied to the 3-element set $\{x_0, x_1, x_2\}$ gives a new neighbour y_2 for one of them. If we are in *Case I*, with y_2 unmatched by M, then there will be a path from y_2 back to x_0 such that

(i) the path has odd length

(ii) the edges alternate between being in M and outside M

(iii) the first and last edges are outside M, since the endpoints of the path are unmatched by M.

Then M^* is obtained from M by replacing the edges along this path inside of M with those outside of M.

If we are in *Case II* then y_2 is matched to a vertex x_3 which, by the same reasoning as before, must be different from the previous vertices. Then we keep going. Each round of this procedure ends up locating a new vertex in X and, since X is a finite set, the procedure must end. When it does, we are in *Case I*, and there will be a path back to x_0 satisfying conditions (i)-(iii) above. M^* is then obtained by exchanging edges along this path as before. This completes the proof of Hall's theorem.

Definition 32 : Let M be a matching in any graph G. A path in G is said to be *M-alternating* or *M-augmenting* if conditions (i)-(iii) above are satisfied by it.

The idea of the above proof is thus that, given a non-perfect matching

M for X, we can find an M-alternating path and obtain a bigger matching by exchanging edges along this path. Since any standard search procedure can be used to locate an M-alternating path, the proof yields an efficient recursive algorithm for constructing a perfect matching in a bipartite graph satisfying Hall's condition.

The cool thing is that the same idea works to find a matching of maximum size in any graph whatsoever :

Theorem 32 *Let G be any graph and M any matching in G which is not maximal in size. Then there exists an M-alternating path in G.*

PROOF : Omitted. See Theorem 17.5.2 in Biggs. Note that Biggs only talks about bipartite graphs, but the exact same proof works for any graph.

This theorem has two consequences :

(A) Call a matching M in a graph *maximal* if there is no matching M^* which properly contains M as a subset. Trivially, every matching of maximum size is maximal, but the converse is not obvious, not to me at least. Theorem 32 tells us it is so.

(B) We can complete part (iii) of the program outlined above, i.e.: we can describe an efficient procedure for constructing a matching of maximum size in any graph whatsoever. Such a method proceeds recursively by taking a matching M as input and searching (via any standard search algorithm) for an M-alternating path. If no such path is located, then M is already of maximum size, by Theorem 32. Otherwise, we replace M by the larger matching M^* obtained by exchanging edges along the alternating path, and repeat the procedure.

Finally, we complete part (ii) of our plan by showing how the theory of matchings for bipartite graphs can be recast in terms of network flows. So let $B = (X, Y, E)$ be a bipartite graph. We construct a corresponding network N_B as follows :

(a) The vertices of N_B are those of B along with two additional vertices s and t.
(b) the edges of N_B are those of B plus : one edge from s to each vertex of X, and one edge from each vertex of Y to t.

(c) the edges of N_B are directed as follows : all edges from s are directed outwards, all edges in B are directed from X to Y, and all edges to t are directed inwards.
(d) every edge in the network gets capacity one.

It is now pretty simple to observe that a maximal flow for this network corresponds to a maximal matching in B. Note that the way the Ford-Fulkerson algorithm works implies that, for any maximal flow, the flow along each edge will be either 0 or 1. Clearly, then, the edges in B which are saturated by such a flow determine a maximal matching for B. In fact, it is not hard to see that the F-F algorithm for this network is just a reformulation of the alternating path procedure described in the proof of Hall's theorem.

As a last remark, Theorem 30 obviously must yield a corresponding statement for bipartite graphs. To state this nicely, we need one further piece of terminology :

Definition 33 : Let $G = (V, E)$ be any graph. A subset W of V is said to *cover* G if every edge of G has at least one of its endpoints in W.

Theorem 33 (König's Theorem) *In a bipartite graph, the maximum size of a matching equals the minimum size of a covering set of vetices.*

Proof : This is a recasting of Theorem 30 in the language of bipartite graphs, though I don't think it is immediately obvious that a minimal cut in the network corresponds to a minimal covering set of vertices in the graph. The proof of this is left as an exercise to the reader. An interesting historical fact is that König's theorem is considerably older than either Hall's theorem or Theorem 30.

Edge-Coloring of Graphs

Definition 34 : An *edge-coloring* of a graph is an assignment of a color to each edge of the graph such that whenever two edges have a vertex in common, they get different colors. The *edge chromatic number* of a graph is the minimum number of colors needed to edge-color it.

Notation : I am not aware of any standard notation for edge chromatic numbers, so I will use my own notation : $\Phi(G)$ will denote the edge chromatic number of a graph G.

Proposition 34 *For any graph G,*

$$\Phi(G) \geq \Delta(G), \tag{46}$$

where $\Delta(G)$ is as defined in Proposition 23.

Proof : This is obvious.

Far less obvious is the following big result :

Theorem 35 (Vizing 1964, Gupta 1966) *For any graph G, $\Phi(G)$ equals either $\Delta(G)$ or $\Delta(G) + 1$.*

I find this theorem very surprising. On the other hand, since any edge-coloring problem can be formally recast as a vertex-coloring problem (for some other graph, whose vertices are the edges of the original graph), it is not at all surprising that

Theorem 36 *In general, the problem of deciding whether $\Phi(G)$ equals $\Delta(G)$ or $\Delta(G) + 1$ is NP-complete.*

But this leads in turn to another suprise, namely that for bipartite graphs the situation is much simpler :

Theorem 37 *If B is a bipartite graph, then $\Phi(B) = \Delta(B)$.*

Proof : The really satisfying thing about the proof of this theorem is that, not only is it constructive, but it uses the same kind of idea as in the matching problem discussed above.

So let $B = (X, Y, E)$ be a bipartite graph. Suppose we have already colored some number $k < |E(B)|$ of its edges with $\Delta(B)$ colors. We show how to replace this with a coloring of $k + 1$ edges, consisting of the old edges plus one more, without using any additional colors.

Since some edge is assumed to be as yet uncolored, pick any such edge $\{x_0, y_0\}$. We shall extend the coloring to this edge.

At least one of our $\Delta(G)$ colors is not already used on an edge through

x_0 : let $\mathcal{C}_1$ be any such color. Similarly, at least one color is not yet used on an edge through y_0 : let $\mathcal{C}_2$ be such a color.

If $\mathcal{C}_1 = \mathcal{C}_2$ then we can simply color the edge $\{x_0, y_0\}$ in this color and we're done. Otherwise, we may assume that $\mathcal{C}_1 \neq \mathcal{C}_2$. In particular, we may assume that $\mathcal{C}_1$ has already been used to color some edge through y_0. Let x_1 be the other endpoint of this edge. Two cases now arise :

Case I : $\mathcal{C}_2$ hasn't yet been used to color an edge through x_1. Then recolor the edge $\{x_1, y_0\}$ with $\mathcal{C}_2$ and use $\mathcal{C}_1$ to color $\{x_0, y_0\}$.

Case II : $\mathcal{C}_2$ already used on an edge through x_1. Let y_1 denote the other endpoint of this edge. Note that our previous assumptions imply that y_1 is different from y_0. Now identify two cases again :

(a) If $\mathcal{C}_1$ hasn't yet been used on an edge through y_1, then consider the path

$$y_1 \overset{\mathcal{C}_2}{\rightarrow} x_1 \overset{\mathcal{C}_1}{\rightarrow} y_0 \rightarrow x_0.$$

We can replace this coloring with

$$y_1 \overset{\mathcal{C}_1}{\rightarrow} x_1 \overset{\mathcal{C}_2}{\rightarrow} y_0 \overset{\mathcal{C}_1}{\rightarrow} x_0,$$

and are done.

(b) If $\mathcal{C}_1$ is already used on an edge through y_1, let x_2 denote the other endpoint of that edge. Previous assumptions imply that x_2 is a different vertex from both x_0 and x_1. Now just repeat the whole procedure.

Since the graph is finite, the procedure must terminate, at which point we will have a path leading back to y_0 of previously colored edges such that colors $\mathcal{C}_1$ and $\mathcal{C}_2$ have been used alternately on these edges, and whichever one of these colors is used on the first edge of the path, the other is not yet used on any other edge through the starting vertex. Then we can extend our coloring to the edge $\{x_0, y_0\}$ by switching colors along this path as far as y_0 and then using $\mathcal{C}_1$ to color $\{x_0, y_0\}$. This completes the proof of the theorem.

Lecture 11 : Tuesday 10/10

We now begin the final part of the course, which is a (very) short introduction to the algebra and arithmetic of discrete structures, a field which has found significant applications in recent times, most notably in the construction of (i) public key cryptosystems for use in secure communication (ii) error-correcting codes for digital transmission over noisy channels. We will just about have the time to develop enough theory to be able to describe one classic application, namely the RSA public key cryptosystem. The reader interested in seeing further applications will need to take more courses !

Let's start off with something very concrete, namely the integers $\mathbf{Z}$. Two so-called *binary operations*[14] can be performed on integers, namely addition (+) and multiplication (×). The following properties are then satisfied :

(I) Addition is *associative*, i.e.: for all $a, b, c \in \mathbf{Z}$ we have

$$(a + b) + c = a + (b + c).$$

(II) There is an *additive identity* or *zero element*, denoted 0, such that, for any $a \in \mathbf{Z}$,

$$a + 0 = 0 + a = a.$$

(III) There exist *additive inverses*, i.e.: for every $a \in \mathbf{Z}$ there exists $b \in \mathbf{Z}$ such that

$$a + b = b + a = 0.$$

Obviously, one usually writes $b = -a$.

(IV) Addition is *commutative*, i.e.: for all $a, b \in \mathbf{Z}$ we have

$$a + b = b + a.$$

(V) Multiplication is associative, i.e.: for all $a, b, c \in \mathbf{Z}$ we have

$$a \times (b \times c) = (a \times b) \times c.$$

[14] Formally, a *binary operation* on a set S is a function from $S \times S \to S$. In other words, it assigns to each ordered pair of elements of S a third element of S.

(VI) There is a *multiplicative identity* or *unit element*, denoted 1, such that, for any $a \in \mathbf{Z}$,

$$a \times 1 = 1 \times a = a.$$

(VII) Multiplication is commutative, i.e.: for all $a, b \in \mathbf{Z}$,

$$a \times b = b \times a.$$

(VIII) Multiplication is *distributive* over addition, i.e.: for all $a, b, c \in \mathbf{Z}$ we have

$$a \times (b + c) = (a \times b) + (a \times c).$$

DEFINITION 35 : Let S be any set on which a binary operation, denoted $+$, has been defined. If (I)-(III) above are satisfied (with $\mathbf{Z}$ replaced by S naturally), we say that $(S, +)$ is a *group*. If in addition (IV) is satisfied, we say that the group is *commutative* or *abelian*.

DEFINITION 36 : Let S be any set on which two binary operations, denoted $+$ and $\times$, have been defined. If (I)-(VI), together with (VIII), are satisfied, we say that $(S, +, \times)$ is a *ring*[15]. If in addition (VII) is satisfied, the ring is said to be *commutative*.

One thing you can't do within the set of integers is divide, so that the analog of (III) for multiplication doesn't hold. But this problem is solved by just extending $\mathbf{Z}$ to $\mathbf{Q}$. In $\mathbf{Q}$ we thus have

(IX) There exist *multiplicative inverses*, i.e.: for any nonzero element $a \in \mathbf{Q}$, there exists a nonzero $b \in \mathbf{Q}$ such that

$$a \times b = b \times a = 1.$$

One usually writes $b = a^{-1}$ or $b = 1/a$.

DEFINITION 37 : If the triple $(S, +, \times)$ satisfies all of (I)-(XI), other than perhaps (VII), then it is called a *division ring*. If (VII) is also satisfied, it is called a *field*.

[15] In some books, the term *ring* will be used so long as only (I)-(V), plus (VIII), are satisfied. If then (VI) is also satisfied, it will be stated that the ring has a *unity*.

So **Z** is the simplest and most natural example of a commutative ring and **Q** is the simplest example of a field. However both these sets are infinite. For the applications we have in mind we will be interested in *finite* groups, rings and fields.

Example 22 : The simplest non-trivial example of a finite field is a set consisting of just two elements, denoted 0 and 1, in which addition is defined as XOR addition, i.e.:

$$0+0=1+1=0, \quad 0+1=1+0=1,$$

and multiplication is defined as the AND operation, i.e.:

$$0\times 0=0\times 1=1\times 0=0, \quad 1\times 1=1.$$

One readily checks that all of properties (I)-(IX) are satisfied. This field is denoted $\mathbf{Z}_2$.

Example 23 : In any ring $(S,+,\times)$ it holds that $s\times 0=0$ for all $s\in S$.

Proof : Let $y:=s\times 0$. Axiom (II) implies in particular that $0+0=0$. Thus, by Axiom (VIII),

$$y=s\times 0=s\times(0+0)=(s\times 0)+(s\times 0)=y+y,$$

i.e.:

$$y=y+y. \tag{47}$$

Next, Axiom (III) says that there exists an element $z\in S$ such that $y+z=0$. Add z to both sides of (47) and we get that

$$0=y+z=(y+y)+z.$$

But by Axiom (I) the right-hand side can be rewritten as $y+(y+z)=y+0=y$, by Axiom (II). Thus $0=y$ as claimed.

Now we shall present a generalisation of Example 22. Several definitions are necessary :

Notation : If x and y are integers, we write $x|y$ to denote that y is a multiple of x.

Definition 38 : Let n be a positive integer, and let a, b be any two integers. We say that a is *congruent* to b *modulo* n, and write $a \equiv b \pmod{n}$, if $n|(a-b)$.

Proposition 38 *Given a positive integer n, the following properties are satisfied by the relation on $\mathbf{Z}$ of congruence modulo n :*

(i) Reflexivity : $a \equiv a$ (mod n) for any a.

(ii) Symmetry : If $a \equiv b$ (mod n) then $b \equiv a$ (mod n).

(iii) Transitivity : If $a \equiv b$ (mod n) and $b \equiv c$ (mod n) then $a \equiv c$ (mod n).

Proof : Exercise.

Definition 39 : A relation on a set S which is reflexive, symmetric and transitive is called an *equivalence relation.* If two elements are related, they are said to be *equivalent.* Given an element $s \in S$, the *equivalence class* of S is the subset of S consisting of all elements equivalent to s.

Proposition 39 *(i) Given an equivalence relation on a set S, any two equivalence classes are either identical or disjoint subsets of S. Hence the equivalence classes partition S.*

(ii) Given a positive integer n, the relation of congruence modulo n partitions the integers into n equivalence classes. Each integer is equivalent to exactly one of the numbers $0, 1, 2, ..., n-1$.

Proof : (i) Exercise. (ii) Each integer is congruent modulo n to its own remainder upon division by n.

Part (ii) of the proposition suggests natural addition $\oplus$ and multiplication $\otimes$ operations on the set $\{0, 1, ..., n-1\}$, namely

$$a \oplus b := \text{the unique } c \in \{0,1,...,n-1\} \text{ such that } a+b \equiv c \pmod{n},$$
$$a \otimes b := \text{the unique } c \in \{0,1,...,n-1\} \text{ such that } a \cdot b \equiv c \pmod{n}.$$

Our claim is that these definitions make the set $\{0, 1, ..., n-1\}$ into a ring. This ring is denoted $\mathbf{Z}_n$, which one should note coincides with the definition in Example 22 when $n = 2$. To verify the claim, the only thing one really needs to check is that the addition and multiplication operations are *well-defined*, by which one means the following :

Proposition 40 *Let n be a positive integer and a, b, c, d arbitrary integers. Suppose $a \equiv b \ (mod \ n)$ and $c \equiv d \ (mod \ n)$. Then $a + c \equiv b + d \ (mod \ n)$ and $a \cdot c \equiv b \cdot d \ (mod \ n)$.*

PROOF : Exercise.

EXAMPLE 24 : If today is Tuesday, what day of the week will it be in 10^{100} days from now ?

Solution : We are interested in the remainder when 10^{100} is divided by 7. Repeated application of Prop. 40 justifies the following sequence of computations, in which all congruences are modulo 7 :

$$10 \equiv 3 \Rightarrow 10^3 \equiv 3^3 = 27 \equiv -1.$$

Thus

$$10^{100} = (10^3)^{33} \cdot 10 \equiv (-1)^{33} \cdot 3 = -3 \equiv 4.$$

In other words, 10^{100} leaves a remainder of 4 when divided by 7 and so in 10^{100} days time it will be a Saturday.

The question which will lead us to somewhat deeper results is :

'*For which n is $\mathbf{Z}_n$ a field ?*'

The next lecture will be occupied with proving the following two theorems :

Theorem 41 *(i) Let n be a positive integer and $a \in \mathbf{Z}_n$. Then a has a multiplicative inverse if and only if $SGD(a, n) = 1$.*

(ii) In particular, $\mathbf{Z}_n$ is a field if and only if n is a prime.

Theorem 42 (Fermat/Euler) *(i) Let n be a positive integer and a any integer such that $SGD(a, n) = 1$. Then*

$$a^{\phi(n)} \equiv 1 \ (mod \ n). \tag{48}$$

(ii) In particular, if n is a prime and a is any integer which is not a multiple of n, then

$$a^{n-1} \equiv 1 \ (mod\ n). \tag{49}$$

Lecture 12 : Friday 13/10

Proof of Theorem 41 : Part (ii) obviously follows from part (i), so we just need to prove the latter. Let $a \in \mathbf{Z}_n$. Then a is invertible if and only if there exists an integer b such that

$$ab \equiv 1 \ (\text{mod } n).$$

This means that n evenly divides $ab - 1$, hence that there exists an integer k such that

$$ab - 1 = kn.$$

Let's change notation a bit : $b \rightarrow x$, $n \rightarrow b$, $k \rightarrow -y$. Then a is invertible if and only if there exist integers x, y such that

$$ax + by = 1.$$

Theorem 41 is thus a special case of part (i) of the following more general result :

Theorem 43 (Euclid's Lemma) *(i) Let a, b, c be integers. Then the equation*

$$ax + by = c \tag{50}$$

has an integer solution (x, y) if and only if $SGD(a, b)$ divides c.

(ii) Let $d := SGD(a, b)$ and let x_0, y_0 be any integers satisfying

$$ax_0 + by_0 = d. \tag{51}$$

Suppose $c = md$. Then the most general integer solution to (50) is

$$x = mx_0 + n\left(\frac{b}{d}\right), \tag{52}$$

$$y = my_0 - n\left(\frac{a}{d}\right), \tag{53}$$

where n is an arbitrary integer. In particular, whenever (50) has at least one integer solution, then it has infinitely many of them.

PROOF OF THEOREM 43, PART I : If $d = \mathrm{SGD}(a,b)$ does not divide c then clearly (50) can have no integer solutions, as the left-hand side is a multiple of d for any choice of x and y. Now suppose d divides c. Clearly it suffices to show that (50) has a solution when $c = d$. The proof that this is the case is constructive, i.e.: we can describe an efficient procedure for locating a solution to (51). The procedure is called *Euclid's algorithm* and is perhaps best illustrated by an example :

EXAMPLE 25 : Find an integer solution to

$$101x + 37y = 1. \tag{54}$$

Solution : Observe that $\mathrm{SGD}(101, 37) = 1$ since both numbers are prime. Thus (54) should have a solution. Euclid's algorithm consists of two steps. The first step performs a sequence of divisions which produces the number $1 = \mathrm{SGD}(101, 37)$, as follows :

$$\begin{aligned} 101 &= 2 \cdot 37 + 27, \\ 37 &= 1 \cdot 27 + 10, \\ 27 &= 2 \cdot 10 + 7, \\ 10 &= 1 \cdot 7 + 3, \\ 7 &= 2 \cdot 3 + 1, \\ 3 &= 3 \cdot 1 + 0. \end{aligned}$$

The procedure terminates once a zero remainder is obtained and the last non-zero remainder is the SGD of the two input numbers. The second step of the algorithm proceeds backwards through the above equations as follows :

$$\begin{aligned} 1 &= 7 - 2 \cdot 3 \\ &= 7 - 2 \cdot (10 - 7) \\ &= 3 \cdot 7 - 2 \cdot 10 \\ &= 3 \cdot (27 - 2 \cdot 10) - 2 \cdot 10 \\ &= 3 \cdot 27 - 8 \cdot 10 \\ &= 3 \cdot 27 - 8 \cdot (37 - 27) \end{aligned}$$

$$
\begin{aligned}
&= 11 \cdot 27 - 8 \cdot 37 \\
&= 11 \cdot (101 - 2 \cdot 37) - 8 \cdot 37 \\
&= 11 \cdot 101 - 30 \cdot 37.
\end{aligned}
$$

This implies that a solution to (54) is given by $x_0 = 11$, $y_0 = -30$.

Assuming that Euclid's algorithm always works, this completes the proof of part (i) of Theorem 53, and thus also the proof of Theorem 51. Let us thus briefly explain why the algorithm works :

The first step is the important one. One needs to know that the last non-zero remainder is always the SGD of the two input numbers. This follows from the following two observations :

(a) Let $d = \text{SGD}(a, b)$ as usual. Then d divides both a and b, so will also divide any integer which can be expressed as an integer-linear combination of a and b. The remainder upon dividing b into a is such an integer. Call this remainder r_1. Thus d divides both b and r_1. The algorithm then proceeds to divide r_1 into b and produce a remainder r_2. By the same reasoning, d divides r_2. And so on ... If the last non-zero remainder is r_k, then we've shown that r_k must be divisible by d.

(b) It thus suffices to show that r_k itself divides both a and b. To see this, one goes backwards through the steps of the algorithm, applying the same reasoning as before, only in reverse. That the $(k+1)$:st remainder is zero means that r_k divides r_{k-1} evenly. But r_k was the result of dividing r_{k-1} into r_{k-2}. Thus r_k divides r_{k-2} evenly. And so on ... we finally arrive at the desired conclusion that r_k divides both b and a evenly.

Proof of Theorem 43, part II : It is clear that, for any $n \in \mathbf{Z}$, if x and y are given by (52) and (53) resp., then (50) is satisfied. It remains to show there are no further solutions. This involves showing the following : given any two solutions (x_1, y_1) and (x_2, y_2) to (50), there exists an integer n such that

$$x_2 = x_1 + n \left(\frac{b}{d} \right), \tag{55}$$

$$y_2 = y_1 - n \left(\frac{a}{d} \right). \tag{56}$$

OK, well we're assuming that

$$ax_1 + by_1 = c, \tag{57}$$
$$ax_2 + by_2 = c. \tag{58}$$

Subtract (57) from (58) to get

$$a(x_2 - x_1) = b(y_1 - y_2).$$

The number d is a common divisor of both sides of this equation. Dividing it out, we get

$$\left(\frac{a}{d}\right)(x_2 - x_1) = \left(\frac{b}{d}\right)(y_1 - y_2). \tag{59}$$

Now the integer b/d divides the left-hand side of (59), but by definition of d, we have $\text{SGD}(a/d, b/d) = 1$. The conclusion we wish to draw from this is that b/d already divides $x_2 - x_1$. Note that if this is so, then (55) follows immediately, and then (56) follows in turn from (59). That our conclusion is indeed valid is a consequence of the following general fact :

Proposition 44 *Let n_1, n_2, n_3 be integers. Suppose that $n_1 | n_2 n_3$ and that $SGD(n_1, n_2) = 1$. Then $n_1 | n_3$.*

To prove this proposition we need in fact a much deeper result, which for good reason is referred to as

Theorem 45 (Fundamental Theorem of Arithmetic) *Every positive integer can be expressed in EXACTLY ONE WAY as a product of primes.*

Theorem 45 is something which probably all of you know, but most of you do not appreciate, due to the abominable way arithmeitc is usually taught in schools !! Anyway, I'm not going to prove it here - the proof is non-trivial, though not hard.

I will just comment on how it immediately implies Prop. 44. That n_1 divdes $n_2 n_3$ means that the entire prime factorisation of the former appears amongst the factorisation of the latter. That $\text{SGD}(n_1, n_2) = 1$ means, on the other hand, that none of the prime factors of n_1 appear amongst the factors of n_2. But the factorsation of $n_2 n_3$ is just the composition of the factorisations of n_2 and n_3. This means that the entire factorisation of n_1 must appear amongst the factors of n_3, hence that n_1 divides n_3, as required.

This completes the proof of Prop. 44, and with it that of Theorem 43(ii).

Proof of Theorem 42 : When n is a prime, then $\mathrm{SGD}(a, n) = 1$ if and only if $\mathrm{SGD}(a, n) \neq n$, i.e.: if and only if a is not a multiple of n. In addition, $\phi(n) = n - 1$ when n is prime. Thus part (ii) of the theorem follows from part (i), and it remains to prove the latter.

Theorem 41 says that $\mathrm{SGD}(a, n) = 1$ if and only if a is invertible, when viewed as an element of $\mathbf{Z}_n$, i.e.: if and only if there exists an integer b such that $ab \equiv 1 \pmod{n}$. Note that this implies that if $\mathrm{SGD}(x, n) = \mathrm{SGD}(y, n) = 1$ then $\mathrm{SGD}(xy, n) = 1$, since if $xx' \equiv yy' \equiv 1 \pmod{n}$, then $(xy)(x'y') \equiv (xx')(yy') \equiv 1 \pmod{n}$ so xy is also invertible in $\mathbf{Z}_n$[16].

Now let $x_1, ..., x_{\phi(n)}$ be all the invertible elements in $\mathbf{Z}_n$. By the above observation, for each $i = 1, ..., \phi(n)$, ax_i is also invertible mod n, hence the numbers $ax_1, ..., ax_{\phi(n)}$ are just a permutation, mod n, of the numbers $x_1, ..., x_{\phi(n)}$. In particular,

$$\prod_{i=1}^{\phi(n)} x_i \equiv \prod_{i=1}^{\phi(n)} ax_i \equiv a^{\phi(n)} \cdot \left(\prod_{i=1}^{\phi(n)} x_i \right) \pmod{n}. \tag{60}$$

Let X denote the prodct on both sides of (60). Since X is a product of invertible elements in $\mathbf{Z}_n$ the same is true, by our observation above, of X itself. Let Y be its inverse mod n. Multiplying both sides of (60) by Y yields (48) and proves Theorem 41.

[16] The same conclusion can be drawn from the Fundamental Theorem of Arithmetic.