BLURB

The text below is virtually identical to that in last year's document. The only change of note is that the Ford-Fulkerson algorithm in network graphs has been removed as we didn't get to that this time. There are several questions on old exams asking you to implement this : you may ignore these exercises. Otherwise, the course material is pretty much identical to last year's. Whereas on the homeworks I have mainly given problems which require some independent thought, the exam is mainly focused on checking that you've learned basic techniques. This will be sufficient to be able to achieve a clear passing grade - for a top grade, you'll need to do a bit more. This situation is far from ideal, but the fact that most of you are still part of the (thankfully soon to be extinct) system where you take 4-5 three point courses simoultaneously makes anything else practically impossible.

END OF BLURB

I have divided the course material into three levels below. You will pass (fail) the exam if you can solve all (none) of the problems from Level 1. There will be 6 questions on the exam, of which about 4 will belong to Level 1, and about one each from each of levels 2 and 3. Note that, by contrast, the nomework exercises have been divided much more evenly between all three levels, with perhaps even a surplus of Level 2 and 3 exercises.

Finally note that, of the 6 questions, 2 will come from Del 1 (Kombinatorik), 2 from Del 2 (Aritmetik), 1 from Del 3 (Grafteori) and 1 will be a 'Wild Card', i.e.: it can come from anywhere or from several places.

Level 1

Everything which involved carrying out some *algorithmic procedure.*

DEL 1 :

Solving linear recurrence relations

$$\begin{aligned} au_{n+2} + bu_{n+1} + cu_n &= f(n), \\ u_0 = \alpha, \qquad u_1 &= \beta. \end{aligned}$$

DEL 2 :

(i) Euclid's algorithm to compute $\mathrm{SGD}(a, b)$.
(ii) Euclid's algorithm to solve linear Diophantine equations

$$ax + by = c,$$

or, equivalently, to solve one-variable linear congruences $ax \equiv b \pmod{n}$.
(iii) Euclid's algorithm to compute $a^{-1} \pmod{n}$ when $\mathrm{SGD}(a, n) = 1$.
(iv) Computing $a^b \pmod{n}$ quickly when a and b are large. The square-and-multiply method always works, but can be slow. If b is much bigger than n and $\mathrm{SGD}(a, n) = 1$, then we can use Fermat's/Euler's theorem.
(v) The Chinese Remainder Theorem to solve a system of linear congruences.
(vi) Solving quadratic congruences in $\mathbf{Z}_p$, where p is a prime.

DEL 3 :

(i) Checking whether a graph has an Euler cycle or path and finding one using a greedy search when it has.
(ii) FInding a Hamilton path or cycle in a graph - there is no good algorithm here, you just have to experiment.
(iii) Greedy algorithm for graph coloring ; computing $\chi(G)$.
(iv) Finding a minimal spanning tree in a weighted graph (Kruskal or Prim algorithm).
(v) Finding a shortest path between two points in a (directed) graph - Dijkstra's algorithm..

Level 2

Some kind of *formula* is available, but it's not perhaps immediately obvious how to use it.

DEL 1 :

(i) Use of the multiplication principle and applications of it (permutations, combinations, binomial theorem, unordered selection with repitition) for counting.
(ii) Inclusion-Exclusion principle.

DEL 2 : Nothing really.

DEL 3 :

(i) Using Euler's formula for plane graphs.

Level 3

Solving the problem requires a bit more *creativity*, or a deeper *understanding* of the theory.

DEL 1 :

(i) Proving a combinatorial identity (usually involving binomial coefficients).
(ii) Finding or verifying (with proof) a recurrence relation.
(iii) Catalan numbers, Stirling numbers, partitions.

DEL 2 :

Everything not included under Level 1. In particular, clever use of the Fundamental Theorem of Arithmetic or of congruences for studying non-linear Diophantine equations.

DEL 3 : We didn't do very much graph theory, so there's not a lot I can say here. We did enough so that I could formulate a challenging problem : make sure you understand the basic definitions, and do a few exercises from the end of the chapters in Grimaldi or Biggs.