Eighteenth Lecture: 16/5

Theorem 18.1. *Prim's algorithm always produces a minimum spanning tree.*

Proof. Let $G = (V, E)$ be a connected, weighted graph on n vertices and let T be a spanning tree for G produced by Prim's algorithm. Let $e_1, e_2, \ldots, e_{n-1}$ be the sequence of edges chosen by the algorithm in order. Let U be any other spanning tree for G and let i be the smallest index such that e_i is not an edge in U. We will show that there is another spanning tree U^* such that $w(U^*) \leq w(U)$ and U^* contains each of the edges $e_1, \ldots, e_i$. Iterating this procedure at most $n - 1$ times will thus show that $w(T) \leq w(U)$ and hence that T is a minimum spanning tree, since U was chosen arbitrarily.

Now U contains each of the edges $e_1, \ldots, e_{i-1}$ by assumption. Let S be the set of vertices spanned by these edges. Let $e_i = \{x, y\}$, where $x \in S$ and $y \in V \backslash S$ - Prim's algorithm always chooses the next edge so that it covers a new vertex. Since U spans G, there must be *some* simple path in U from x to y. Call this path $\mathcal{P}_{xy}$. This path starts at a vertex in S and ends at a vertex in $V \backslash S$. Hence there must be a first edge $e : z \to w$ along the path such that $z \in S$ and $w \in V \backslash S$. Now the edge e was available to Prim's algorithm at the i:th step but was not chosen ahead of e_i. Since the algorithm always chooses an edge of minimal weight amongst those available, we must have $w(e) \geq w(e_i)$. Let $U^* := (U \cup \{e_i\}) \backslash \{e\}$. Thus $w(U^*) \leq w(U)$ and U^* contains all the edges $e_1, \ldots, e_i$. So all that remains to be shown is that U^* is a spanning tree for G. Since U^* has the same number of edges as the spanning tree U, it suffices to show that it spans G, for then it will follow immediately from Theorem 17.4(iv) that it contains no cycles.

So let $v_1, v_2 \in V$. Since U spans G, there is a unique simple path in U from v_1 to v_2. Call it $\mathcal{P}_{v_1 v_2}$. If this path doesn't use the edge e then it is still present in U^*. So suppose the path $\mathcal{P}_{v_1 v_2}$ *does* use the edge e. Let $\mathcal{C}_{xy}$ be the simple cycle formed by the path $\mathcal{P}_{xy}$ above and the edge e_i. Then we obtain a path in U^* from v_1 to v_2 by

- following the path $\mathcal{P}_{v_1 v_2}$ until we hit the edge e,
- then replacing e by the rest of the cycle $\mathcal{C}_{xy}$, traversed in the appropriate direction,
- finally continuing to y along the path $\mathcal{P}_{v_1 v_2}$.

We've proven that U^* contains a path between any pair of vertices of G, hence it spans G, v.s.v. □

Remark 18.2. The proof that Kruskal's algorithm also produces a MST is very similar to this and is left as an exercise to the reader. See also Exercises 16.7.9 and 16.7.10 on the homepage.

Definitions 18.3. Let $G = (V, E)$ be a graph. A *matching* in G is a subset $M \subseteq E$ such that no two edges of M share a vertex. In other words, M is a subgraph of G in which every vertex has degree zero or one. A vertex of degree one is said to be *included* in the matching, or simply to be *matched*.

The *size* of a matching M is the number of edges in it, and is denoted $|M|$. If $|M| \geq |M'|$ for any other matching M', then M is said to be a *maximum matching*.

A matching is said to be *perfect* or *complete* if $|M| = |V|/2$, in other words, if every vertex in G is matched.

Remark 18.4. Note the simple but useful observation that a graph cannot possess a perfect matching if it has an odd number of vertices.

The *Matching Problem* asks for a procedure to determine a maximum matching in a graph and thus, in particular, to decide if the graph has a perfect matching. The problem makes most sense in the setting of bipartite graphs $G = (X, Y, E)$. Among the myriad interpretations of the problem in this setting, here are three of the most common ones:

A. X is a set of men and Y a set of women. An edge represents a pair such that each regards the other as an "acceptable" spouse. Hence, a maximum matching represents an optimal solution to the *Marriage Problem* of marrying off as many couples as possible.

B. X is a set of job seekers and Y a set of vacancies. An edge represents a job for which the corresponding person is qualified. Hence, a maximum matching represents an optimal solution to the *Job Assignment Problem* of getting jobs for as many people as possible.

C. X is a set of high-school leavers and Y a set of university degree programs. An edge represents a program which the corresponding student is both qualified for and interested in. Hence, a maximum matching represents a way of getting the maximum number of students into university.

Terminology 18.5. In a bipartite graph $G = (X, Y, E)$, the size of any matching cannot exceed $\min\{|X|, |Y|\}$. In particular, there cannot exist a perfect matching if $|X| \neq |Y|$. If $|X| \leq |Y|$ (resp. if $|Y| \leq |X|$) and there exists a matching of size $|X|$ (resp. of size $|Y|$), then this matching is said to be *perfect for X, or X-perfect (resp. perfect for Y, or Y-perfect).*

Definition 18.6. Let $G = (V, E)$ be any graph and $v \in V$. The *neighborhood* of v, denoted $N(v)$, is the set of all vertices joined to v by an edge, i.e.:

$$N(v) = \{w \in V : \{v, w\} \in E\}.$$

More generally, if $A \subseteq V$, the neighborhood of A, denoted $N(A)$, is the union of the neighborhoods of its elements, i.e.: $N(A) = \cup_{v \in A} N(v)$.

Theorem 18.7. (Hall's Marriage Theorem) *Let $G = (X, Y, E)$ be a bipartite graph. Then there exists a perfect matching for X if and only if*

$$|N(A)| \geq |A| \quad \forall\, A \subseteq X. \tag{18.1}$$

Note that (18.1) is called *Hall's condition*. Before proving the theorem, we need to introduce the central idea in the proof:

Definition 18.8. Let M be a matching in a graph. A simple path $v_1 v_2 \ldots v_k$ in G is called an *M-augmenting path* if

(i) k is even, i.e.: the length of the path is odd
(ii) every second edge in the path lies in M and every second edge lies outside M
(iii) the first edge lies outside M, i.e.: $\{v_1, v_2\} \notin M$.
(iv) v_1 and v_k are unmatched vertices in M.

One immediately observes the following: Let M be a matching and $\mathcal{P}$ an M-augmenting path, Let $M' \subseteq E$ be the set of edges obtained from M by replacing the edges along $\mathcal{P}$ which lie in M by those which don't. Then M' is also a matching and $|M'| = |M| + 1$. Hence, if there exists an M-augmenting path, M is not a maximum matching.

Proof of Hall"s theorem. It is obvious that Hall's condition is necessary since, if $A \subseteq X$ and $x \in A$, then x can a priori only be matched with a vertex in $N(A)$. In an X-perfect matching, every vertex of A must be matched, so there must be at least as many vertices in $N(A)$ as there are in A.

So suppose Hall's condition is satisfied. Let M be any matching such that $|M| < |X|$. It suffices to prove the existence of an M-augmenting path. Since M is not X-perfect, there is at least one unmatched node in X. Pick one, call it x_0. Set $A := \{x_0\}$. Hall's condition says $|N(A)| \geq |A| = 1$, so x_0 has at least one neighbor. Pick a neighbor, call it y_0. If y_0 is unmatched, then $x_0 y_0$ is an M-augmenting path.

So we may suppose y_0 is matched, say to x_1. Set $A := \{x_0, x_1\}$. Hall's condition says $|N(A)| \geq |A| = 2$, so there is at least one more vertex, other than y_0, which is a neighbor of either x_0 or x_1. Pick such a vertex, call it y_1. If y_1 is unmatched and a neighbor of x_0, then $x_0 y_1$ is an M-augmenting path. If y_1 is unmatched and a neighbor of x_1, then $x_0 y_0 x_1 y_1$ is an M-augmenting path.

So we may assume that y_1 is matched, say to x_2. Keep iterating the above procedure to produce a sequence of distinct vertices $x_0, y_0, x_1, y_1, x_2, y_2, \ldots, x_k, y_k$, until you hit an unmatched vertex y_k. Note that this must eventually happen since applying Hall's condition with $A = X$ implies that $|Y| \geq |X|$ and hence, if there is an unmatched vertex in X there must also be one in Y. Once we hit an unmatched y_k, there will be a shortest path from y_k back to x_0 which only passes through vertices among those in the above sequence, and such that every second edge along this path is of the form $\{x_k, y_{k-1}\}$ and included in M. Hence every other edge is not in M, since M is a matching. The path starts in Y and ends in X, so it has odd length. The first and last vertices, y_k and x_0, are unmatched. Hence this is an M-augmenting path, v.s.v. $\square$

Crucially, the above proof immediately yields an algorithm for finding a maximum matching in a bipartite graph, namely:

Augmenting path algorithm for bipartite graphs. Start with the empty matching $M = \phi$. Perform a breadth-first search for an M-augmenting path. If no such path is found, conclude that M is a maximum matching and stop. If such a path is found, replace M by the augmented matching M', got by exchanging edges along the M-augmenting path. Repeat.