Discrete Mathematics Lent 2009
MA 210

Notes for lectures 15 and 16

3.8 Trees

A *tree* is a connected graph with no cycles.

Theorem 3.13. *Every tree on n vertices has $n-1$ edges.*

A *forest* is a graph with no cycles. Every component of a forest is a tree.

Let G be a graph on n vertices. A *spanning tree* in G is a set of edges of G which form a tree on n vertices.

Lemma 3.14. *A graph contains a spanning tree if and only if the graph is connected.*

3.9 Minimal spanning trees

Suppose we have a collection of cities which have to be connected by a cable network. For each pair of cities u and v we have the following options: either it is impossible to construct a cable between u and v, or it is possible to construct a cable between u and v and we know the cost c_{uv} for constructing this cable. We would like to find a collection of cables such that each city is connected with every other city (not necessarily by a direct cable) and such that the cost of the total network is as small as possible.

We model this problem in the following way: form a graph G whose vertex set $V(G)$ is the set of cities and the edge set is

$$E(G) = \{uv \mid \text{ a cable between } u \text{ and } v \text{ can be constructed}\}.$$

Now we want to find a subset of edges of $E(G)$ that form a connected graph on the set of cities. It is obvious that, if we want to minimize the total cost, this subgraph should not contain any cycles and, therefore, it should be a spanning tree of G.

So the question can be rephrased to: find a spanning tree T of G such that $\sum_{uv\in E(T)} c_{uv}$ is minimal.

Such a tree is called a *minimal spanning tree* or a *spanning tree with minimum total cost*.

Theorem 3.15 (Kruskal's Algorithm)**.** *Let G be a connected graph on n vertices, with a cost value c_e for each $e \in E(G)$. Then a minimal spanning tree of G can be found as follows:*

(i) Let e_1 be the edge with minimum cost, i.e.,

$$c_{e_1} = \min_{e\in E(G)} c_e.$$

Set $W = \emptyset$.

(ii) *Obtain* $e_2, e_3, \ldots, e_{n-1}$ *recursively: having found* $e_1, e_2, \ldots, e_k$, *let* f *be the edge with minimum weight in* $E(G) \setminus (\{e_1, e_2, \ldots, e_k\} \cup W)$.

If we can form a cycle with edges from $e_1, e_2, \ldots, e_k, f$, *then add* f *to* W *and repeat this step.*

If we cannot form a cycle with edges from $e_1, e_2, \ldots, e_k, f$, *then set* $e_{k+1} = f$ *and repeat this step as long as* $k + 1 < n - 1$.