DISCRETE MATHEMATICS

W W L CHEN

Chapter 18

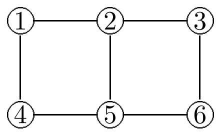
WEIGHTED GRAPHS

18.1. Introduction

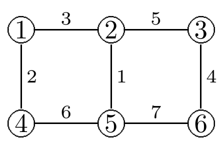
We shall consider the problem of spanning trees when the edges of a connected graph have weights. To do this, we first consider weighted graphs.

DEFINITION. Suppose that $G = (V, E)$ is a graph. Then any function of the type $w : E \to \mathbb{N}$ is called a weight function. The graph G, together with the function $w : E \to \mathbb{N}$, is called a weighted graph.

EXAMPLE 18.1.1. Consider the connected graph described by the following picture.



Then the weighted graph



has weight function $w : E \to \mathbb{N}$, where

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

and

$$w(\{1, 2\}) = 3, \qquad w(\{1, 4\}) = 2, \qquad w(\{2, 3\}) = 5, \qquad w(\{2, 5\}) = 1,$$
$$w(\{3, 6\}) = 4, \qquad w(\{4, 5\}) = 6, \qquad w(\{5, 6\}) = 7.$$

DEFINITION. Suppose that $G = (V, E)$, together with a weight function $w : E \to \mathbb{N}$, forms a weighted graph. Suppose further that G is connected, and that T is a spanning tree of G. Then the value

$$w(T) = \sum_{e \in T} w(e),$$

the sum of the weights of the edges in T, is called the weight of the spanning tree T.

18.2. Minimal Spanning Tree

Clearly, for any spanning tree T of G, we have $w(T) \in \mathbb{N}$. Also, it is clear that there are only finitely many spanning trees T of G. It follows that there must be one such spanning tree T where the value $w(T)$ is smallest among all the spanning trees of G.

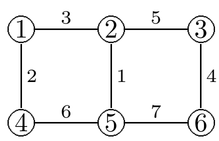
DEFINITION. Suppose that $G = (V, E)$, together with a weight function $w : E \to \mathbb{N}$, forms a weighted graph. Suppose further that G is connected. Then a spanning tree T of G, for which the weight $w(T)$ is smallest among all spanning trees of G, is called a minimal spanning tree of G.

REMARK. A minimal spanning tree of a weighted connected graph may not be unique. Consider, for example, a connected graph all of whose edges have the same weight. Then every spanning tree is minimal.

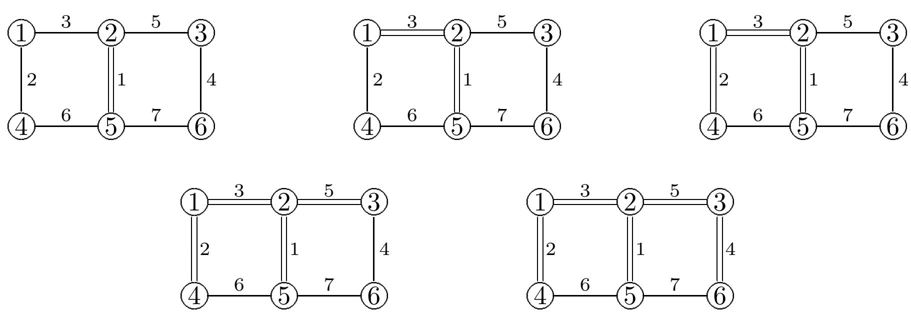
The question now is, given a weighted connected graph, how we may "grow" a minimal spanning tree. It turns out that the Greedy algorithm for a spanning tree, modified in a natural way, gives the answer.

PRIM'S ALGORITHM FOR A MINIMAL SPANNING TREE. *Suppose that $G = (V, E)$ is a connected graph, and that $w : E \to \mathbb{N}$ is a weight function.*
(1) Take any vertex in V as an initial partial tree.
(2) Choose edges in E one at a time so that each new edge has minimal weight and joins a new vertex in V to the partial tree.
(3) Stop when all vertices in V are in the partial tree.

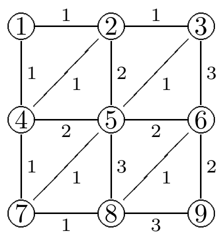
EXAMPLE 18.2.1. Consider the weighted connected graph described by the following picture.



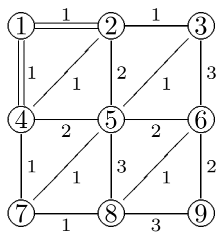
Let us start with vertex 5. Then we must choose the edges $\{2, 5\}, \{1, 2\}, \{1, 4\}, \{2, 3\}, \{3, 6\}$ successively to obtain the following partial spanning trees, the last of which represents a minimal spanning tree.



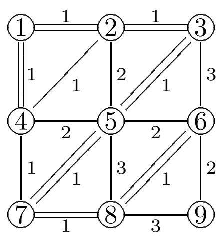
EXAMPLE 18.2.2. Consider the weighted connected graph described by the following picture.



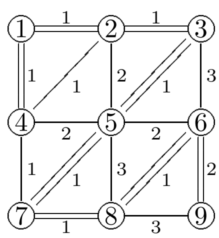
Let us start with vertex 1. Then we may choose the edges $\{1,2\},\{1,4\}$ successively to obtain the following partial tree.



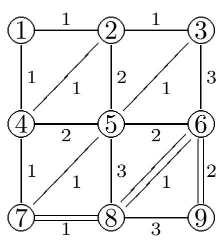
At this stage, we may choose the next edge from among the edges $\{2,3\}$ and $\{4,7\}$, but not the edge $\{2,4\}$, since both the vertices 2 and 4 are already in the partial tree, so that we would not be joining a new vertex to the partial tree but would form a cycle instead. From this point, we may choose $\{2,3\}$ as the third edge, followed by $\{3,5\},\{5,7\},\{7,8\},\{6,8\}$ successively to obtain the following partial tree.



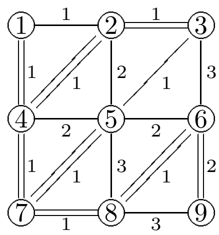
At this point, we are forced to choose the edge $\{6,9\}$ to complete the process. We therefore obtain the following minimal spanning tree.



However, if we start with vertex 9, then we are forced to choose the edges $\{6,9\},\{6,8\},\{7,8\}$ successively to obtain the following partial tree.



From this point, we may choose the edges $\{4,7\},\{1,4\},\{2,4\},\{5,7\},\{2,3\}$ successively to obtain the following different minimal spanning tree.



PROPOSITION 18A. *Prim's algorithm for a minimal spanning tree always works.*

PROOF. Suppose that T is a spanning tree of G constructed by Prim's algorithm. We shall show that $w(T) \leq w(U)$ for any spanning tree U of G. Suppose that the edges of T in the order of construction are $e_1, \ldots, e_{n-1}$, where $|V| = n$. If $U = T$, clearly the result holds. We therefore assume, without loss of generality, that $U \neq T$, so that T contains an edge which is not in U. Suppose that

$$e_1, \ldots, e_{k-1} \in U \qquad \text{and} \qquad e_k \notin U.$$

Denote by S the set of vertices of the partial tree made up of the edges $e_1, \ldots, e_{k-1}$, and let $e_k = \{x, y\}$, where $x \in S$ and $y \in V \setminus S$. Since U is a spanning tree of G, it follows that there is a path in U from x to y which must consist of an edge $\overline{e}$ with one vertex in S and the other vertex in $V \setminus S$. In view of the algorithm, we must have $w(e_k) \leq w(\overline{e})$, for otherwise the edge $\overline{e}$ would have been chosen ahead of e_k in the construction of T by the algorithm. Let us now remove the edge $\overline{e}$ from U and replace it by e_k. We then obtain a new spanning tree U_1 of G, where

$$w(U_1) = w(U) - w(\overline{e}) + w(e_k) \leq w(U).$$

Furthermore,

$$e_1, \ldots, e_{k-1}, e_k \in U_1.$$

If $U_1 \neq T$, then we repeat this process and obtain a sequence of spanning trees $U_1, U_2, \ldots$ of G, each of which contains a longer initial segment of the sequence $e_1, e_2, \ldots, e_{n-1}$ among its elements than its predecessor. It follows that this process must end with a spanning tree $U_m = T$. Clearly

$$w(U) \geq w(U_1) \geq w(U_2) \geq \ldots \geq w(U_m) = w(T)$$

as required. ○

If Prim was lucky, Kruskal was excessively lucky. The following algorithm may not even contain a partial tree at some intermediate stage of the process.

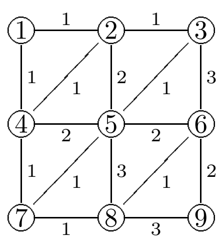
KRUSKAL'S ALGORITHM FOR A MINIMAL SPANNING TREE. *Suppose that $G = (V, E)$ is a connected graph, and that $w : E \to \mathbb{N}$ is a weight function.*

(1) Choose any edge in E with minimal weight.

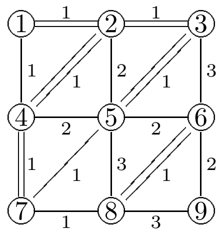
(2) Choose edges in E one at a time so that each new edge has minimal weight and does not give rise to a cycle.

(3) Stop when no more edges can be chosen.

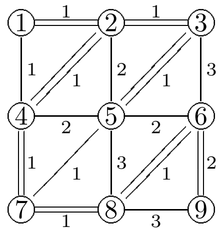
EXAMPLE 18.2.3. Consider again the weighted connected graph described by the following picture.



Choosing the edges $\{1,2\},\{3,5\},\{6,8\},\{2,4\},\{4,7\},\{2,3\}$ successively, we arrive at the situation below.



We must next choose the edge $\{7,8\}$, for choosing either $\{1,4\}$ or $\{5,7\}$ would result in a cycle. Finally, we must choose the edge $\{6,9\}$. We therefore obtain the following minimal spanning tree.



PROPOSITION 18B. *Kruskal's algorithm for a minimal spanning tree always works.*

PROOF. Suppose that T is a spanning tree of G constructed by Kruskal's algorithm, and that the edges of T in the order of construction are $e_1,\ldots,e_{n-1}$, where $|V|=n$. Let U be any minimal spanning tree of G. If $U=T$, clearly the result holds. We therefore assume, without loss of generality, that $U\neq T$, so that T contains an edge which is not in U. Suppose that

$$e_1,\ldots,e_{k-1}\in U \qquad \text{and} \qquad e_k\notin U.$$

Let us add the edge e_k to U. Then this will produce a cycle. If we now remove a different edge $\overline{e}\notin T$ from this cycle, we shall recover a spanning tree U_1. In view of the algorithm, we must have $w(e_k)\leq w(\overline{e})$, for otherwise the edge $\overline{e}$ would have been chosen ahead of the edge e_k in the construction of T by the algorithm. It follows that the new spanning tree U_1 satisfies

$$w(U_1)=w(U)-w(\overline{e})+w(e_k)\leq w(U).$$

Since U is a minimal spanning tree of G, it follows that $w(U_1)=w(U)$, so that U_1 is also a minimal spanning tree of G. Furthermore,

$$e_1,\ldots,e_{k-1},e_k\in U_1.$$

If $U_1\neq T$, then we repeat this process and obtain a sequence of minimal spanning trees $U_1,U_2,\ldots$ of G, each of which contains a longer initial segment of the sequence $e_1,e_2,\ldots,e_{n-1}$ among its elements than its predecessor. It follows that this process must end with a minimal spanning tree $U_m=T$. ◯

18.3. Erdős Numbers

Every mathematician has an Erdős number. Every self-respecting mathematician knows the value of his or her Erdős number.

The idea of the Erdős number originates from the many friends and colleagues of the great Hungarian mathematician Paul Erdős (1913–1996), who wrote something like 1500 mathematical papers of the highest quality, the majority of them under joint authorship with colleagues. The Erdős number is, in some sense, a measure of good taste, where a high Erdős number represents extraordinarily poor taste.

All of Erdős's friends agree that only Erdős qualifies to have the Erdős number 0. The Erdős number of any other mathematician is either a positive integer or ∞. Since Erdős has contributed more than anybody else to discrete mathematics, it is appropriate that the notion of Erdős number is defined in terms of graph theory.

Consider a graph $G = (V, E)$, where V is the finite set of all mathematicians, present or "departed". For any two such mathematicians $x, y \in V$, let $\{x, y\} \in E$ if and only if x and y have a mathematical paper under joint authorship with each other, possibly with other mathematicians. The graph G is now endowed with a weight function $w : E \to \mathbb{N}$. Since Erdős dislikes all injustices, this weight function is defined by writing $w(\{x, y\}) = 1$ for every edge $\{x, y\} \in E$.

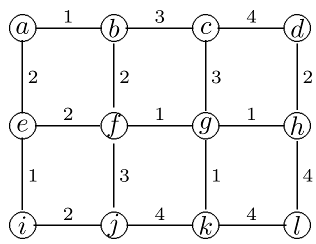
If we look at this graph G carefully, then it is not difficult to realize that this graph is not connected. Well, some mathematicians do not have joint papers with other mathematicians, and others do not write anything at all. It follows that there are many mathematicians who are in a different component in the graph G to that containing Erdős. These mathematicians all have Erdős number ∞.

It remains to define the Erdős numbers of all the mathematicians who are fortunate enough to be in the same component of the graph G as Erdős. Let $x \neq$ Erdős be such a mathematician. We then find the length of the shortest path from the vertex representing Erdős to the vertex x. The length of this path, clearly a positive integer, but most importantly finite, is taken to be the Erdős number of the mathematician x.

It follows that every mathematician who has written a joint paper with Uncle Paul, by which Erdős continues to be affectionately known, has Erdős number 1. Any mathematician who has written a joint paper with another mathematician who has written a joint paper with Erdős, but who has himself or herself not written a joint paper with Erdős, has Erdős number 2. And so it goes on!

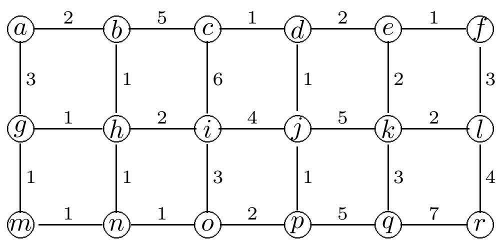
Problems for Chapter 18

1. Use Prim's algorithm to find a minimal spanning tree of the weighted graph below.



2. Use Kruskal's algorithm to find a minimal spanning tree of the weighted graph in Question 1.

3. Use Prim's algorithm to find a minimal spanning tree of the weighted graph below.



4. Use Kruskal's algorithm to find a minimal spanning tree of the weighted graph in Question 3.