CSC 101 - Spring 2013
Lab 5 - Networks

1. Our first problem in this lab relates to connecting a number of buildings (i.e., a campus) with fiber optic cable. Each building will have a router and wireless network equipment which will serve a local area network within the building. Routers can be configured to route network traffic from any building to any other building **provided** there is some fiber optic cable path from the source to the destination. It is Ok if the network traffic has has to go through several routers to get to its destination. That's good news: we do not have to directly connect every building to every other building. **But**, we have to provide enough connections so there is a pathway from every building to every other.

   The cost of connecting different pairs of buildings varies, depending on the distance between the buildings, and depending on what lies between the buildings. For example, walkways that are cut-through by a trench digger must be rebuilt, adding to the cost. Cutting through a parking lot, and making subsequent repairs is even more expensive.

   It may be impractical to directly connect certain pairs of buildings. For these buildings, every plausible trench plan either 1) goes though another building which is in the way, or 2) crosses natural gas pipelines which is not allowed due to safety regulations, or 3) goes through property on which we have no legal right-of-way. For these pairs of buildings, we will designate the cost of connecting these buildings as $+\infty$ to indicate that they can not be directly connected. You have sent your field engineers out to survey the site, and obtain contract bids on the cost of connecting each pair of buildings. We label our eight buildings: B_1, B_2, B_3, B_4, B_5, B_6, B_7, and B_8. The report of the field engineers indicates the following costs (in thousands of dollars):

	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
B_1	0	12	$+\infty$	$+\infty$	15	$+\infty$	$+\infty$	$+\infty$
B_2	12	0	10	4	2	7	$+\infty$	$+\infty$
B_3	$+\infty$	10	0	6	$+\infty$	8	$+\infty$	$+\infty$
B_4	$+\infty$	4	6	0	$+\infty$	$+\infty$	$+\infty$	5
B_5	15	2	$+\infty$	$+\infty$	0	11	9	17
B_6	$+\infty$	7	8	$+\infty$	11	0	3	13
B_7	$+\infty$	$+\infty$	$+\infty$	$+\infty$	9	3	0	$+\infty$
B_8	$+\infty$	$+\infty$	$+\infty$	5	17	13	$+\infty$	0

   Finally, it is your job to ensure that the project is completed at the lowest possible cost. You must come up with an infrastructure plan (a list of pairs of buildings which will be directly connected) which is lowest cost of all possible infrastructure plans. You must also convince the Comptroller's office and the Financial V.P. that your plan is optimal.

   The business problem described above is accurately modeled by a weighted graph. A weighted graph is defined as follows:

Definition: A *weighted graph* is a tuple $G = (V, E)$ where V is a set of n vertices

$$V = \{ v_1, v_2, v_3, ..., v_n \}$$

and a set of m edges

$$E = \{ (v_{i_1}, v_{j_1}), (v_{i_2}, v_{j_2}), (v_{i_3}, v_{j_3}), \dots, (v_{i_m}, v_{j_m}) \} \quad ,$$

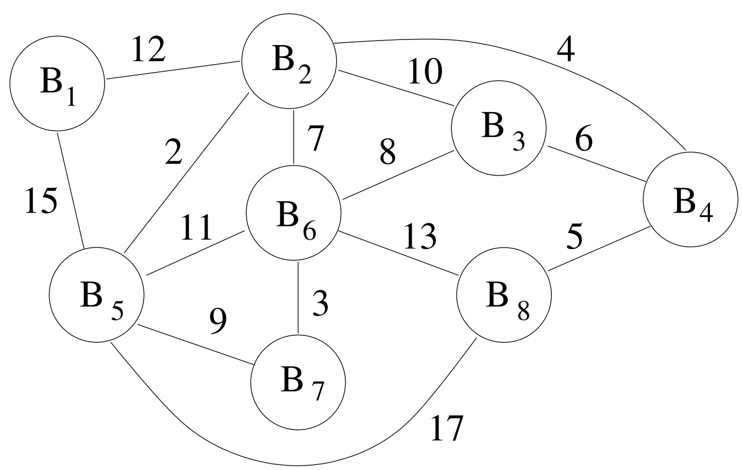
together with a cost function

$$f : V \times V \rightarrow (\mathcal{R}^+ \cup +\infty) \quad .$$

A sub-graph (with no cycles) which connects all of the vertices in the graph is called a **spanning tree**.

For a weighted graph, we can define the **cost** of a spanning tree to be the sum the costs of all of the edges in the graph. Various spanning trees are possible, and each will have a (possibly) different cost. A **minimal spanning tree** is a spanning tree which is as low cost as any other spanning tree. An important theorem relating to minimal spanning trees is: *If all of the edge costs are distinct, then the minimal spanning tree is unique.*[1]

The cost table shown above can be illustrated by the following weighted graph:



Fortunately, there are several straightforward algorithms to solve the minimal spanning tree problem. One such algorithm is known as Kruskal's algorithm[2].

There are several approaches to algorithm design. Kruskal's algorithm takes an approach known as a **greedy algorithm**. A greedy algorithm makes a "greedy" local choice without any global knowledge; further, a sequence of greedy local choices leads to a globally correct solution. In the case of Kruskal's algorithm, the local decision is to choose the next lowest-cost edge. I.e., if you want a lowest-cost spanning tree, build it using the lowest-cost edges. At the point in time where the algorithm makes the

[1] The curious reader is referred to the graph theory literature for a mathematical proof of this theorem.

[2] Kruskal's algorithm was first published in the Proceedings of the American Mathematical Society in 1956. An alternative is Jarnik's algorithm (a.k.a. Prim's algorithm), first published by V. Jarnik in Czech in 1930, and later independently published by R. C. Prim in 1957.

"greedy" choice, it has not examined the other edges (i.e., it has no "global" knowledge of the problem details). Not all problems can be solved by a greedy approach.[3] Mathematical proof is required to show that the sequence of greedy local choices leads to a globally correct solution.

A pseudo-code statement of Kruskal's algorithm is given below:

Input: A weighted undirected graph $G = (V, E)$ with weight function $f : V \times V \rightarrow \mathcal{R}^+ \cup +\infty$

Output: A minimal spanning tree T for G.

Method:

> $T = \phi$; /* Empty set. */
> $S = \phi$; /* Empty set. */
> Construct a priority queue Q of all edges in E ;
> **for** each vertex $v \in V$ **do** add $\{v\}$ to S.
> **while** $||S|| > 1$ **do**
>   choose edge (v, w) from Q of lowest cost ;
>   delete (v, w) from Q ;
>   **if** (v and w are in different sets W_1 and W_2 in S) **then**
>   Replace W_1 and W_2 in S by $W_1 \cup W_2$;
>   Add (v, w) to T ;
>   **end if**
> **end while**

Your Task: Use Kruskal's algorithm to find a solution to the network planning problem posed above. Draw a separate diagram to illustrate the steps in the algorithm as each edge is added to the minimal spanning tree. As you consider the edges in order of increasing cost, indicate which edges are not added to the minimal spanning tree.

What is the total cost of your fiber optic infrastructure plan ? Are you sure there does not exist a lower cost plan (why or why not) ?

Your Next Task: Use Dijkstra's algorithm to find the shortest path from B_1 to all the other vertices. On each step, indicate the vertex that you choose, and the current contents of the array D.

A pseudo-code statement of Dijkstra's algorithm is given below:

Input: A weighted undirected graph $G = (V, E)$ with weight function $f : V \times V \rightarrow \mathcal{R}^+ \cup +\infty$
A source vertex v_1.

Output: A table D indicating cost of the shortest path from v_1 to every other vertex in V.

[3] For example, a greedy algorithm approach fails miserably at solving the well known "Knapsack Problem". E.g., $\{3, 5, 7, 15, 19, 22, 30\}$ with $K = 26$.

Method:

```
S = { v1 }    /* Set of vertices for which shortest path is known. */

for each vertex v do
    D[v] = f(v1, v)    /* Cost of the edge from v1 to v. */

while ( S ≠ V ) do
    choose vertex w not in S such that D[w] is smallest
    add w to S
    for each vertex v in V but not in S do
        D[v] = min( D[v] , D[w] + f(w, v) )
    end for
end while
```