

**PROGRAMMING ASSIGNMENT 3**

DUE: Wednesday November 22, 7 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read <http://www.student.cs.uwaterloo.ca/~cs341> for general instructions and policies.

1. [20 marks] Implement Algorithm **A** from Assignment 8. Input to the algorithm is an undirected graph  $G = (V, E)$  and an integer weight  $w(e)$  for each edge  $e$  with  $w(e) > 0$ . The algorithm depends on a real-valued parameter  $t \geq 1$ . Let the graph produced by the algorithm be called  $H_t$ .

**Algorithm A**

```

sort  $E$  such that  $w(e_1) \leq w(e_2) \leq \dots \leq w(e_m)$ 
 $H \leftarrow \phi$ 
for  $i = 1 \dots m$ 
    let  $e_i = (u, v)$ 
    compute  $d_H(u, v)$  in the graph with edge set  $H$ 
    if  $t \cdot w(e_i) < d_H(u, v)$  then add  $e_i$  to  $H$ 
end
```

You will need a shortest path algorithm to compute  $d_H(u, v)$ . You may use Bellman-Ford—it is easier to implement than Dijkstra since there is no need for a priority queue.

**Input:**

The first line is a single positive integer  $n$ , the number of vertices. Vertices are numbered  $1, \dots, n$ . The second line is a single positive integer  $m$ , the number of edges. The third line is a single real number  $t \geq 1$ . Following that are  $m$  lines giving the edges. Each edge is given as three whitespace delimited numbers, the first two giving the endpoint vertices of the edge and the third giving the integer weight of the edge. For example, an edge  $(5, 6)$  of weight 3 would appear as:

```
5 6 3
```

or as:

```
6 5 3
```

**Output:**

The first line is the number of edges of  $H_t$ , call it  $m'$ . The second line is the sum of the weights of the edges of  $H_t$ . Following that are  $m'$  lines giving the edges of  $H_t$ , where the endpoints of each edge are listed in increasing order and the edges are sorted by first vertex and in the case of equal first vertex, sorted by the second vertex.

For example, for the following input:

```
4
```

```
5
1.5
1 2 5
2 3 6
4 1 12
3 4 9
3 1 15
```

the output is:

```
4
32
1 2
1 4
2 3
3 4
```

OPTIONAL [no marks]. One case of practical importance is when the vertices are points in the plane, and the graph is the complete graph with edge weights equal to Euclidean distances. In this case it is known that the number of edges in  $H_t$  is  $O(\frac{n}{t-1})$ . This means that, for fixed  $t > 1$ ,  $H_t$  has  $O(n)$  edges.

In more general cases,  $H_t$  may have  $\Theta(n^2)$  edges—for example in the case of the complete graph with edge weights of 1 and  $t = 1.5$ ,  $H_t$  is the complete graph.

Experiment with points in the plane to check that the number of edges is  $O(\frac{n}{t-1})$ , and to try to find the constant inside the big-oh. Note that you will need to modify your code to deal with real-valued weights on the edges.