

ASSIGNMENT 8

DUE: Wednesday November 15, 7 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read <http://www.student.cs.uwaterloo.ca/~cs341> for general instructions and policies. In particular, note that “giving” an algorithm includes justifying correctness and run time.

1. [10 marks] Use the shortest path algorithms you learned in class to give efficient algorithms for the following problems. Clearly indicate which algorithms you are using. As always, give pseudocode and argue correctness (you may assume correctness of the algorithms presented in class). Clearly state and justify the run time, and try to make your algorithm as fast as possible.
 - (a) [5 marks] Find the weight of a minimum weight cycle in a directed graph. The graph may have negative weight edges but has no negative weight cycle.
 - (b) [5 marks] Find the weight of a maximum weight path in a directed acyclic graph.. The graph has weights on the edges.
2. [20 marks] For any graph $G = (V, E)$ with positive edge weights, the minimum spanning tree T is the “cheapest” subgraph (both in terms of total weight and in terms of number of edges) that connects all the vertices. However, two vertices that are “close” in G may be “far apart” in T .

More precisely, let $d_G(u, v)$ be the length of a shortest path between u and v in G . Let H be any subgraph of G formed by deleting some edges, and let $d_H(u, v)$ be the length of a shortest path between u and v in H . Note that $d_H(u, v) \geq d_G(u, v)$. Consider the worst case ratio of $d_H(u, v)$ to $d_G(u, v)$:

$$\alpha(H) = \max \left\{ \frac{d_H(u, v)}{d_G(u, v)} : u, v \text{ vertices of } G \right\}.$$

- (a) [4 marks] Show that if T is the minimum spanning tree of G then $\alpha(T)$ can be linear in n , the number of vertices. In particular, describe for any n , an example of an edge-weighted graph on n vertices and a pair of vertices u and v in the graph s.t. $d_T(u, v) = \Theta(nd_G(u, v))$.

For the purpose of bounding α , there are alternatives to the minimum spanning tree. Consider the following generalization of Kruskal’s greedy minimum spanning tree algorithm. Input to the algorithm is a graph $G = (V, E)$ and a 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
```

- (b) [5 marks] Prove that H_t always includes the edges of a minimum spanning tree. You may assume that edge weights are unique to make this easier.
- (c) [5 marks] Prove that $\alpha(H_t) \leq t$.
- (d) [6 marks] In terms of n , the number of vertices, and m , the number of edges of G , analyze the running time of algorithm **A** if you compute $d_H(u, v)$ using:
 - i. Dijkstra's algorithm
 - ii. Bellman-Ford

For your 3rd programming assignment you will be asked to implement algorithm **A**. Details available soon.