# CS130N Problem set 8: Weighted graphs

April 24, 2001

1. Discuss with your friends the similarity of the proofs of correctness of Dijkstra's, Kruskal's and Prim's algorithms. Abstract out the main steps of proof of correctness of any greedy algorithm.

2. Reconstruct the proof of correctness of the Bellman-Ford algorithm discussed in the class.

3. Given an example of an $n$-vertex graph $G$ that causes Dijkstra's algorithm to run in $\Omega(n^2 \log n)$ time.

4. Give an example of a weighted directed graph $G$ with negative-weight edges, but no negative-wight cycle, such that Dijkstra's algorithm incorrectly computes the shortest path distance.

5. How is the Bellman-Ford algorithm able to work with negative weights?

6. In Dijkstra's algorithm, when we add a new weight $z$ to $C$, let $w$ be a node not in $C$. Is it possible that the new shortest special path from the source to $w$ should pass first by $z$ and then by some other node of $C$?

7. What can you say about the time required by Kruskal's algorithm if, instead of providing a list of edges, the user supplies a matrix of distances, leaving to algorithm the job of working out which edges exist?

8. Show that if all the weights in a connected graph $G$ are distinct, then there is exactly one minimum spanning tree for $G$.

9. The problem of finding a subset $T$ of the edges of a connected graph $G$ such that all nodes remain connected when only the edges of $T$ are used, and the sum of weights of edges in $T$ is as small as possible, still makes sense even if $G$ has edges with negative weights. However, the

solution may no longer be a tree. Adapt either Kruskal's or Prim's algorithm to work on a graph that may include negative weights.

10. Suppose you are given a timetable, which consists of:

    - a set $\mathcal{A}$ of $N$ airports, and for each airport $a \in \mathcal{A}$, a minimum connect time $c(a)$
    - A set $\mathcal{F}$ of $M$ flights, and the following information, for each flight $f \in \mathcal{F}$:
        - Origin airport $a_1(f) \in \mathcal{A}$
        - Destination airport $a_2(f) \in \mathcal{A}$
        - Departure time $t_1(f)$
        - Arrival time $t_2(f)$

    Give an efficient algorithm for the following problem:

    Given airports $a$ and $b$, and a time $t$, find a sequence of flights that allows one to arrive at the earliest possible time in $b$ when departing from $a$ at or after time $t$. Minimum connecting time at intermediate airports should be observed.

11. Design an efficient algorithm for finding a longest directed path from a vertex $s$ to a vertex $t$ of an acyclic weighted digraph $G$. Specify the graph representation used and any auxiliary data structures used. Analyze the time complexity of your algorithm.