Algorithms and Data Structures
Spring 2018
Exercises for Units 21 and 22

1. Let $G = (V, E, w)$ be a directed, weighted graph with vertex set $V = \{1, \ldots, n\}$ and let D be its distance matrix. I.e. for $1 \leq i, j \leq n$ the entry $D[i, j]$ equals the length of the shortest path from vertex i to vertex j.

   Now assume a graph G' is obtained from G by adding a vertex $n+1$ and weights $w([i, n+1\rangle)$ and $w([n+1, i\rangle)$ for $1 \leq i \leq n$.

   (a) How can you obtain the distance matrix D' of G' from the distance matrix D of G? How much time does this take?

   (b) Does your method work for negative weights also? If yes, how would you deal with negative cycles?

   (c) Can you obtain a general All-Pairs-Shortest-Path algorithm from your approach? If yes, what is its running time?

2. Explicitly representing the shortest paths between all n^2 vertex pairs of an n-vertex graph may take space up to $\Theta(n^3)$ (since $\Theta(n^2)$ of those paths may consists of $\Theta(n)$ edges each).

   Give a representation that uses space just $O(n^2)$ and that allows to report for any give pair u, v of vertices a shortest path π from u to v in time proportional to the number of edges of π.

3. Assume all edge lengths in a graph G are integers in the range 1 through K. Show that in such a graph the single source shortest path problem for source vertex s can be solved in time $O(m + n + K + D_s)$, where D_s is the largest distance from s to any vertex in G. (Of course n is the number of nodes in the graph, and m is the number of edges.)

4. How would you maintain a minimum spanning tree of a graph under insertions and deletions of edges?

   In other words, assume for some undirected, edge-weighted graph $G = (V, E; w)$ a minimum spanning tree T has been computed. Now a new edge e with weight w_e is added to the G and you are supposed to computed the minimum spanning tree of the new graph from T. How would you do this? How much time does it take?

   How would you proceed if an edge were removed from G?