Practice Problems on Dijkstra's Shortest Path Algorithm

CS:3330 Fall 2015

1. Consider the directed edge-weighted graph shown below.
   (Downloaded from `http://siddarthareddy.weebly.com/blog/dijkstras-algorithm-example`).

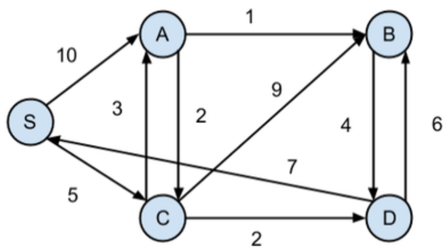

Figure 1: Directed Edge-weighted graph.

   (a) Show the execution of Dijkstra's shortest path algorithm (pseudocode given below) for solving the Single Source Shortest Path (SSSP) problem on this graph. Use the vertex S as the source. For each iteration of the **while**-loop show (i) the vertices in S, (ii) the d'-values assigned to the vertices in $V \setminus S$ during that iteration, and (iii) the vertex v^* selected in that iteration.

> $S \leftarrow \{s\}$; $d[s] \leftarrow 0$
> **while** $S \neq V$ **do**
>   **for** each vertex $u \in V \setminus S$ **do**
> $d'[u] \leftarrow \infty$
>
>   **for** each vertex $u \in V \setminus S$ **do**
> $d'[u] \leftarrow \min_{(v,u) \in E, v \in S}\{d[v] + w(v,u)\}$
>
>   Select a vertex $v^* \in V \setminus S$ with smallest d'-value
> $d[v^*] \leftarrow d'[v^*]$
> $S \leftarrow S \cup \{v^*\}$

   (b) As discussed in class, it is not necessary (and somewhat inefficient) to recompute the d'-values from scratch in every iteration. The efficient way to do this would be to remember d' values from the previous iteration and simply update a few d' values, specifically due to a vertex v^* migrating from $V \setminus S$ to S in each iteration. Show the execution of this modified version of Dijkstra's algorithm. Make sure to identify the edges that were processed in each iteration in order to update d'-values.

   (c) Strictly speaking, the pseudocode given above is not correct. This is because S may never become equal to V since some vertices in the input graph may not be reachable from the source s. In this case, the algorithm will be stuck in an infinite loop. Modify the pseudocode so that it is correct even when not all vertices are reachable from s.

2. Let G be a directed, edge-weighted graph such that every edge has a weight that belongs to the set $\{0, 1, \ldots, W\}$, where W is a non-negative integer. Modify the implementation of Dijkstra's algorithm so that the SSSP problem can be solved in $O(n \cdot W + m)$ time for a graph with n vertices and m edges.