Introduction to Algorithms

chapter 24-25

exercises

> 24.3-2 Give a simple example of a directed graph with negative-weight edges for which Dijkstra's algorithm produces incorrect answers. Why doesn't the proof of Theorem 24.6 go through when negative-weight edges are allowed?

Consider any graph with a negative cycle. RELAX is called a finite number of times but the distance to any vertex on the cycle is $-\infty$, so Dijkstra's algorithm cannot possibly be correct here. The proof of theorem 24.6 doesn't go through because we can no longer guarantee that$\delta(s, y) \le \delta(s, u)$.

> 24.3-3 Suppose we change line 4 of Dijkstra's algorithm to the following.
>
> ```
> 4  while |Q| > 1
> ```
>
> This change causes the **while** loop to execute $|V| - 1$ times instead of $|V|$ times. Is this proposed algorithm correct?

Yes, the algorithm is correct. Let u be the leftover vertex that does not get extracted from the priority queue Q. If u is not reachable from s, then

$u.d = \delta(s, u) = \infty.$

If u is reachable from s, then there is a shortest path

$p = s \to x \to u.$

When the node x was extracted,

$x.d = \delta(s, x)$

and then the edge (x, u) was relaxed; thus,

$u.d = \delta(s, u).$

> 24.3-10 Suppose that we are given a weighted, directed graph $G = (V, E)$ in which edges that leave the source vertex s may have negative weights, all other edge weights are nonnegative, and there are no negative-weight cycles. Argue that Dijkstra's algorithm correctly finds shortest paths from s in this graph.

The proof of correctness, Theorem 24.6, goes through exactly as stated in the text. The key fact was that $\delta(s, y) \le \delta(s, u)$. It is claimed that this holds because there are no negative edge weights, but in fact that is stronger than is needed. This always holds if y occurs on a shortest path from s to u and $y \ne s$ because all edges on the path from y to u have nonnegative weight. If any had negative weight, this would imply that we had "gone back" to an edge incident with s, which implies that a cycle is involved in the path, which would only be the case if it were a negative-weight cycle. However, these are still forbidden.

chapter 34

exercises

> 34.1-2 Give a formal definition for the problem of finding the longest simple cycle in an undirected graph. Give a related decision problem. Give the language corresponding to the decision problem.

The problem $\text{LONGEST-SIMPLE-CYCLE}$ is the relation that associates each instance of a graph with the longest simple cycle contained in that graph. The decision problem is, given k, to determine whether or not the instance graph has a simple cycle of length at least k. If yes, output 1. Otherwise output 0. The language corresponding to the decision problem is the set of all $\langle G, k\rangle$ such that $G = (V, E)$ is an undirected graph, $k \ge 0$ is an integer, and there exists a simple cycle in G consisting of at least k edges.