ADSA
Problem Set 3

1. Give an example of a directed graph of not more than three vertices with edges having negative or positive weights such that Dijkstra will fail to give weights of shortest paths from a source vertex in that graph. Mention the source vertex in your graph and show how Dijkstra will fail to give weights of correct shortest paths.

2. Consider the following algorithm that tackles the issue of negative-weight edges while using Dijkstra to find a shortest path from s to t in directed weighted graph G with some edges having negative weights.

NWD(G, s, t):

1. Find the edge of the least weight, say k, in G. (Note that k will be negative.)
2. Construct a new graph G' which is the same as G except every edge's weight is increased by $-k$. (This makes all the edge weights non-negative.)
3. Find a shortest path P from s to t in G' using Dijkstra.
4. Return the path corresponding to P in G as the answer.

Does **NWD** correctly finds the shortest path in G? Justify your answer.

3. Give a $\theta(|V| + |E|)$ time algorithm to compute shortest path from a source vertex s to every other vertex of a directed acyclic weighted graph where weights are real numbers.

4. Suppose we have a flow network where anti-parallel edges are allowed, that is, it possible to have both (u, v) and (v, u) edges. Suggest a modification in such a flow network so that we can use the max flow computing method for flow networks where anti-parallel edges are not allowed to compute the max flow for those flow networks where anti-parallel edges are allowed.

5. Prove that the Ford-Fulkerson method will terminate when capacities are rational numbers.

6. Suppose that you wish to find, among all the minimum cuts in a flow network G with integer capacities, one that contains the smallest number of edges. Show how to modify the capacities of G to create a new flow network G' in which any minimum cut in G' is a minimum cut with the smallest number of edges in G.

7. In a directed graph G, edge-disjoint s-t paths are a set of paths from vertex s to vertex t such that every pair of paths are edge-disjoint, that is, they share no edge between them. Show how we can find maximum edge-disjoint s-t paths using max-flow computation in a related flow network.

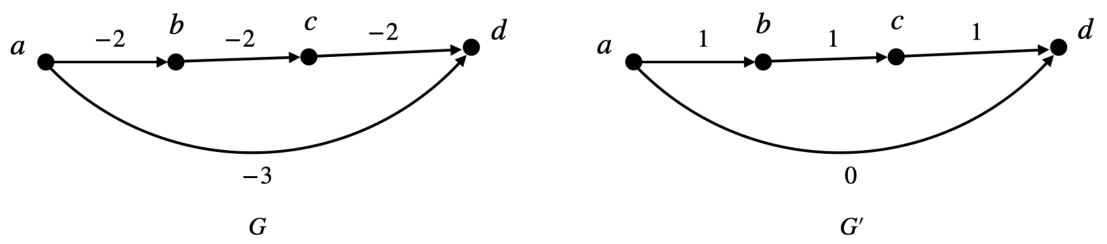
Solutions

Solution 1

Consider a directed graph on three vertices s, t, x with edges: (s, t) with weight 3, (s, x) with weight 4, (x, t) with weight -3. In this graph, with s as the source, Dijkstra will calculate 3 as the weight of the shortest path from s to t, while the right answer is 1.

Solution 2

Algorithm does not work because the path corresponding to the shortest path in G' may not be the shortest path in G. See the below example.



Solution 3

See this.

Solution 4

Suppose the flow network $G = (V, E)$ has some anti-parallel edges. We construct a corresponding network $G' = (V', E')$ which contains the same set of vertices, edges and capacities as G. Then, for every pair of anti-parallel edges (u, v) and (v, u) with capacities c_1 and c_2, respectively, we break the edge (v, u) into two edges (v, w) and (w, u) with both having c_2 as the capacity. Now we can compute max flow $f*$ in G'. Due to flow conservations edges (v, w) and (w, u) must have the same flow, say f. We can easily get a corresponding flow in G, where every edge which was not broken in G' will get the same flow as in G' and any edge (v, u) which was broken into two edges (v, w) and (w, u) in G' will get the the flow of (v, w) and (w, u). This corresponding flow will also be a maximum flow in G.

Solution 5

Multiply all the capacities with the LCM of the denominators of all the capacities. Run Ford-Fulkerson on the modified flow network of integer capacities. We know that Ford-Fulkerson will terminate. Now argue that if it terminates for the modified flow network of integer capacities, it will terminate for the original flow network of rational capacities as well.

Solution 6

See this.

Solution 7

See here.