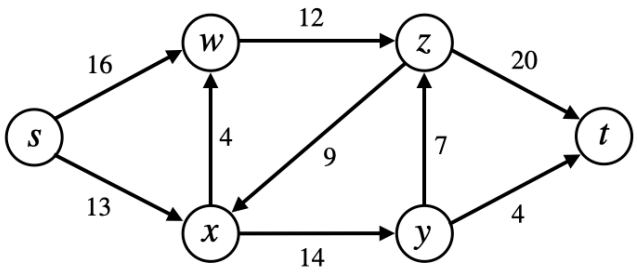
DSAT (August 2025)

Assignment 2

Instructions:

1. *In questions asking for algorithm you should write pseudocode or give detailed explanation in words. In case you are writing algorithms in words, make sure you are not missing necessary details.*
2. *You are allowed to take help but do not copy-paste the answers.*
3. *Doing the problems on your own will help you in exams as extensions of some of these problems may come in exams.*
4. *All questions are of* 5 *marks, but they are not of equal difficulty level.*

1. In the below graph, what is the correct order of vertices dequeued from the priority queue if we want to find the shortest paths from the source vertex s using Dijkstra's algorithm?



2. Let G be a weighted directed graph with integer weights and s be a vertex in G. Prove that if for every vertex v in G there is a shortest path from s that does not contain a negative weight edge, then Dijkstra's algorithm will correctly find the weight of the shortest paths from s to every other vertex.

3. Suppose we have a directed graph $G = (V, E)$ with a designated source vertex s and destination vertex t. We want to find out if there is a path from s to t which has exactly k edges. Recall that a path cannot have repeated vertices.

For any vertex v and for any number i, let us define a Boolean variable $path(v, i)$, which is supposed to be **True** if and only if there is a path from v to t with exactly i edges. As the base case we take $path(t,0)$ to be **True**. We write the following procedure to compute $path(v, i)$ starting from $i = 1$ to k, for every vertex $v \in V$:

For each vertex u such that there is an edge from v to u, we check $path(u, i - 1)$. If any of them is **True** then we set $path(v, i)$ as **True**. Using this logic, we compute $path(v, i)$ for all pairs (v, i).

We finally output $path(s, k)$. Is the described algorithm correct? If yes, justify its correctness. If not, explain why?

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. Casually justify why the proposed modification works.

5. Let $G = (V, E)$ be a flow network with source s, sink t, and integer capacities. Suppose that we are given a maximum flow in G. Suppose that we increase the capacity of a single edge $(u, v) \in E$ by 1. Give an $O(|V| + |E|)$ time algorithm to update the maximum flow. Give a formal justification of correctness and runtime of your algorithm.

6. You are given a directed acyclic graph $G = (V, E)$ with real-valued weights and two distinguished vertices s and t. The weight of a path is the sum of the weights of the edges in the path. Give a dynamic-programming algorithm for finding a longest weighted simple path from s to t. What is the running time of your algorithm? Explain the working of your algorithm.