Graphs I. BFS, Dijkstra, and MST

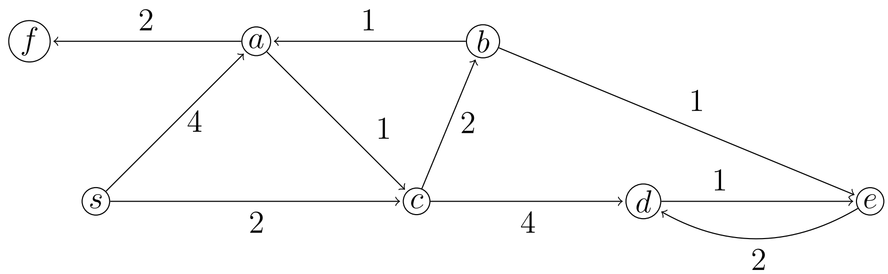

Figure 1: Graph H.

1. 1. Demonstrate a BFS traversal of graph G starting from s, where G is an undirected graph obtained from H by removing edges direction and deletion of parallel edges. Construct the breadth first tree.

2. Find shortest paths in the graph H from the vertex s to all (reachable) vertices via the Dijkstra's algorithm. Demonstrate the algorithm run step by step.

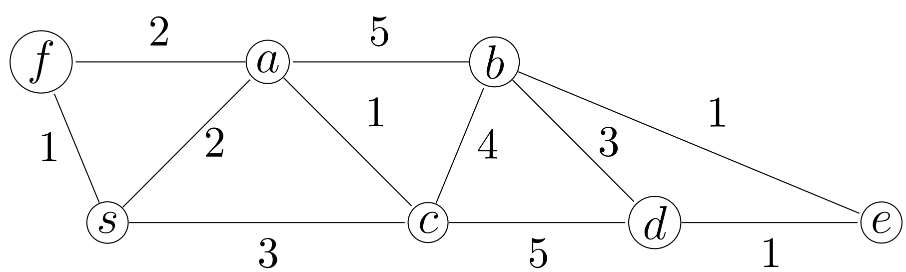

Figure 2: Graph G'.

2. Construct an MST for the graph G'

a) via Kruskal's algorithm;

b) via Prim's algorithm that starts from the vertex s.

3. Prove that if a weighted undirected graph has all distinct (positive) edges' weights, then it has a unique MST.

4. Show how to find the maximum spanning tree of a graph, that is, the spanning tree of largest total weight.

5. The following statements may or may not be correct. In each case, either prove it (if it is correct) or give a counterexample (if it isn't correct). Always assume that the graph $G = (V, E)$ is undirected. Do not assume that edge weights are distinct unless this is specifically stated.

a) If graph G has more than $|V| - 1$ edges, and there is a unique heaviest edge, then this edge cannot be part of a minimum spanning tree.

b) If G has a cycle with a unique heaviest edge e, then e cannot be part of any MST.

c) The shortest-path tree computed by Dijkstra's algorithm is necessarily an MST.

d) Prim's algorithm works correctly when there are negative edges.

e) If one decreases a weight of an edge e that belongs to an MST T, then T remains an MST.

6. The input of the problem is a directed weighted graph with integer weights and its vertices s and t, such that there exists a shortest path from s to t of at most k edges. Construct an algorithm that finds a shortest path from s to t in $O(k|E|)$.

A directed graph is a *tournament* if it can be obtained from a complete undirected graph by setting directions of edges.

7. The adjacency matrix of a tournament $G(V, E)$ is stored in RAM. Construct an $O(|V|)$ algorithm that finds the sink t that is reachable from all the vertices or verifies that such a sink does not exists. (A sink is a vertex of outdegree 0.)

8. Suppose you need to find all shortest paths from the vertex s and it is known that all weights are from the range $0, 1, \ldots, W$, where W is a constant.

1. Modify BFS or Dijkstra's algorithm so that the resulting algorithm solves the problem in $O(W|V| + |E|)$.

2. Propose another algorithm of the complexity $O((|V| + |E|) \log |W|)$.