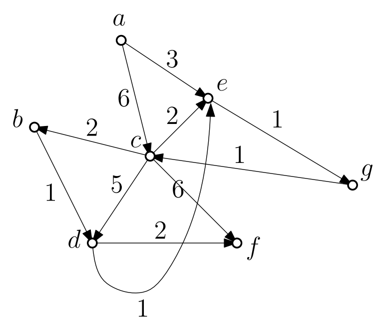
CSCI2100: Special Exercise Set 13

Prepared by Yufei Tao

Problem 1. Consider the weighted directed graph below.



Suppose that we run Dijkstra's algorithm starting from vertex a. Recall that the algorithm relaxes the outgoing edges of every other vertex in turn. Give the order of vertices by which the algorithm relaxes their edges.

Problem 2. Consider a simple directed graph $G = (V, E)$ where each edge $(u, v) \in E$ carries a non-negative weight $w(u, v)$. Given two vertices $u, v \in V$, function $spdist(u, v)$ represents the shortest path distance from u to v. Given a vertex $v \in V$, denote by $\text{IN}(v)$ the set of in-neighbors of v. Let s and t be two distinct vertices in G. Prove:

$$spdist(s, t) = \min_{v \in IN(t)} \{spdist(s, v) + w(v, t)\}.$$

(Hint: First prove LHS $\leq$ RHS, and then prove $\geq$.)

Problem 3. Give a counterexample to show that Dijkstra's algorithm does not work if edge weights can be negative.

Problem 4. Adapt Dijkstra's algorithm to solve the SSSP problem on a weighted undirected graph.

Problem 5. Let $G = (V, E)$ be a weighted directed graph. Give an algorithm to compute the shortest path distances between all pairs of vertices. Your algorithm should finish in $O(|V|(|V| + |E|) \log |V|)$ time.