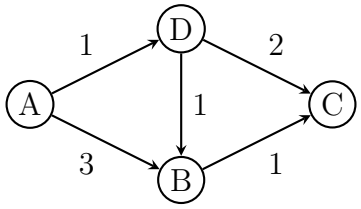
COMS21103: Problems set 5

Shortest paths

Most problem sets for the next few weeks will contain at least one starred problem, which is **optional**, more challenging and hopefully interesting. If any of the problems seem unclear, please post a question on the forum.

1. Let G be the following directed graph.



   Simulate (by hand) the Bellman-Ford algorithm and Dijkstra's algorithm on G with vertex A as the source, and write down the shortest paths output to vertices B, C, D.

2. Imagine we modify the Bellman-Ford algorithm so it stops after only $V - 2$ iterations of the for loop in line 3. Give an example of a graph for which the modified algorithm does not correctly solve the single-source shortest paths problem.

3. In this question you will prove the path-relaxation property stated in lecture, in two parts. Throughout the question, assume that s is the source, $s.d$ is initially set to 0, and $v.d$ is initially set to ∞ for all vertices $v \neq s$.

   (a) Prove the following claim: after any sequence of Relax operations applied to any set of edges, $v.d \geq \delta(s, v)$ for all vertices v. Hint: use induction on the number of Relax operations and the triangle inequality.

   (b) Prove the following claim: If $p = s \to v_1 \to \ldots \to v_k \to v$ is a shortest path from s to v, and the edges in p are relaxed in the order they appear in p (possibly with other edges relaxed in between), then at the end of this process, $v.d \leq \delta(s, v)$. Hint: induction.

   Combining these two claims, we have that $v.d = \delta(s, v)$ after the edges in p are relaxed in order, which is the path-relaxation property.

4. Prove the claim made in lecture that the binary heap operations Parent, Left and Right work correctly as defined.

5. Give an example of a graph G with at least one negative-weight edge such that Dijkstra's algorithm does not correctly output a shortest path in G. Explicitly identify where the property of having only non-negative weight edges was used in the proof of correctness of Dijkstra's algorithm.

6. Imagine we want to "fix" a graph with negative-weight edges to make Dijkstra's algorithm work, and do this by adding some large constant to the weight of each edge to make all the weights positive. Give an example which shows that this approach does not work.

7. ($\star$) Imagine we would like to implement Dijkstra's algorithm using a restricted priority queue which does not support the DecreaseKey operation (so it only supports Insert and Extract-Min). Modify Dijkstra's algorithm to work using such a queue.