MA010 Tutorial 3

Frédéric Dupuis

> *"There is a difference between knowing the path and walking the path."*
>
> Morpheus, The Matrix, 1999

This tutorial covers material from lectures 3 and 4.

Problem 1

In class, we saw that Dijkstra's algorithm can find the shortest path between two nodes if all the edge weights are positive, while the other shortest-path algorithms work even with negative edges.

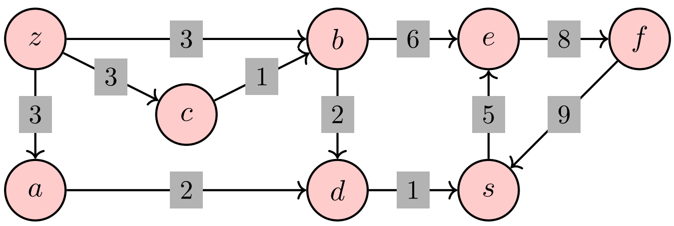
(a) Why this restriction? Find an example of a directed graph with negative edges (but no negative cycle) and a starting vertex for which Dijkstra's algorithm returns the wrong answer.

(b) Simulate the Bellman-Ford algorithm on this graph with the same starting vertex.

(c) Simulate the Floyd-Warshall algorithm on this graph.

Problem 2

Consider the following directed graph with the given flow capacities:



Simulate the Edmonds-Karp algorithm on this graph, with z as the source and s as the sink, to find the maximum flow. What is the minimum cut found by the algorithm? *(Recall that the Edmonds-Karp algorithm is the Ford-Fulkerson algorithm, but where we specify that the search on the residual graph is done using BFS.)*

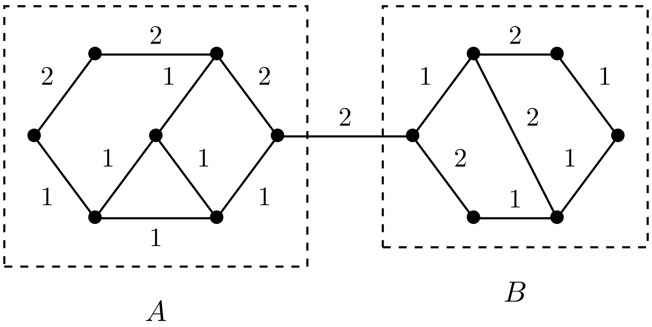
Problem 3

Consider a undirected, weighted graph G with positive weights. Given a path p in G, we define the "bottleneck value" $b(p)$ to be the weight of the edge with the largest weight along p. Given two vertices u and v, we then define $B(u, v)$ to be the minimum bottleneck value over all paths connecting u and v. Suggest an algorithm that computes $B(u, v)$ for every pair of vertices in the graph, and prove that it is correct.

Source: www.cs.arizona.edu/classes/cs445/spring06/hw4.pdf

Problem 4

Consider the following graph:



(a) Find the diameter of the two subgraphs labelled A and B, and the diameter of the whole graph.

(b) Find the edge with the highest reach. Compute its reach, and prove that no other edge in the graph has a higher reach.

Recall the definition of the reach of an edge e:

$$\text{reach}(e) := \max\{\min\left(d(\text{prefix}(p, e)), d(\text{suffix}(p, e))\right) : \forall \text{path } p \in \mathcal{P}_e\}$$

where $\mathcal{P}_e$ is the set of all paths containing e that are a shortest path between their two ends, and where $\text{prefix}(p, e)$ *is the part of p that comes before e, and* $\text{suffix}(p, e)$ *is the part of p that comes after e.*

Problem 5

Let G be a directed, weighted graph with no negative cycles. Suggest an algorithm that finds the weight of a minimum-weight cycle in G.