Problem Set 6

Assigned: June 29
Due: July 6

Problem 1

Construct an algorithm to solve the following problem: Given a graph \( G = (V, E) \), and a subset of vertices, \( K \subseteq V \) compute, for every vertex \( v \in V - K \), the shortest path from the closest vertex in \( K \) to \( V \). For instance, each vertex in \( K \) might be a fire station, and each vertex in \( V - K \) a potential address of a fire. You want to compute, in advance, for every address \( v \), what fire station is nearest to \( v \) and what is the shortest path that a fire truck should take from that station to \( v \). The running time of your algorithm should be \( O(E \cdot \lg(V)) \).

Hint: Modify the graph by adding an additional source vertex that has an arc to every vertex in \( K \).

Problem 2

Someone has proposed the following algorithm to solve the single-source shortest path problem for graphs with negative cost arcs but no negative weight cycles.

function SingleSourceShortestPath(G: directed graph; s: source vertex)
{ 
  \( M = \) the minimum cost of any edge in \( G \);
  if \( (M < 0) \) add \(-M\) to every edge in \( G \).
  since \( G \) now has no negative cost edges,
  apply Dijkstra’s algorithm to solve the shortest path problem from \( s \).
}

Obviously, the result here does not give the correct costs of the shortest paths, but give an example to show that it does not even find the shortest paths.

Problem 3

Someone has proposed the following algorithm to solve the all-pairs shortest path problem in a directed graph.

function AllPairsShortestPath(G: directed graph);
{ for (each vertex \( v \) in \( G \))
  use Dijkstra’s algorithm to solve the single-source shortest path problem from \( v \).
}

Under what circumstances if any does this work better than the Floyd-Warshall algorithm?
Problem 4

Consider the following problem. You are given a directed graph $G$ with two disjoint subsets $A$ and $B$. A path is considered invalid if it goes first through a vertex in $A$ and then through a vertex in $B$. For example, $A$ and $B$ may be points in enemy countries, and $B$ may prohibit travellers whose passport has a visa to $A$. Or in an epidemic of a communicable disease, one may want to block people who have been through $A$ from entering $B$.

A. Modify Dijkstra’s algorithm so that it returns the optimal valid paths.

B. Modify the Floyd-Warshall algorithm so that it returns the optimal valid paths.