# Lab 13

1. Carry out the steps of Dijkstra's algorithm to compute the length of the shortest path between vertex V and vertex Y in the graph below. Your final answer should consist of three elements:

   a) The length of the shortest path from V to Y
   b) The list A[] which shows shortest distances between V and every other vertex
   c) The list B[] which shows shortest paths between V and every other vertex
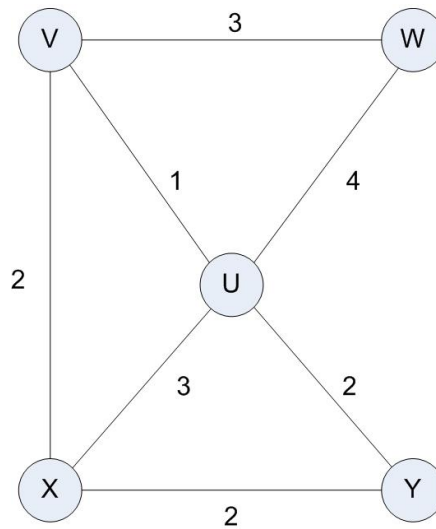

2. *Points about Dijkstra's Algorithm*

   a. What is the shortest path from A to C in the graph below?
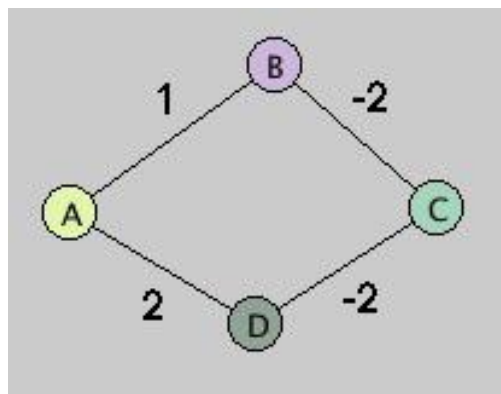
b.

Why is Dijkstra's approach to
the shortest path problem better
than simply using BFS, as
described in the previous
lesson?

↓ BFS Style

[BFS approach: Making all edge weights
= 1 is same as removing all weights.
Perform BFS with start vertex s and
compute distance to each vertex by
returning its *level* in the BFS spanning
tree. These computed values should be
same as values found using Dijkstra]
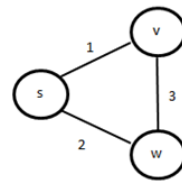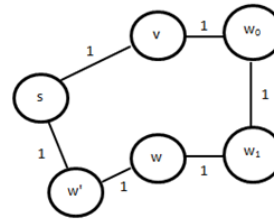
3. Describe an algorithm for deleting a key from a heap-based priority queue that runs in
O(log n) time, where n is the number of nodes. (Hint: You may use auxiliary storage
as the priority queue is built and maintained. Assume there are no two nodes have the
same key.) This technique is needed for the optimized Dijkstra algorithm discussed in
the slides.

Kruskal's Algorithm

4. Carry out the steps of Kruskal's algorithm for the following weighted graph. Keep
track of edges as they are added to T and show the state of evolving clusters through
each iteration of the main while loop.