Lab assignment solution

Problem 1:

Steps

① $X = \{v\}$, $A[v] = 0$, $B[v] = \{\}$

$POOL = \{(v,w), (v,u), (v,x)\}$

$A[w] = A[v] + wt(v,w) = 0+3 = 3$

$A[u] = A[v] + wt(v,u) = 0+1 = 1$ ← min

$A[x] = A[v] + wt(v,x) = 0+2 = 2$

$A[u] = 1$

$B[u] = B[v] \cup \{(v,u)\}$

$= \{(v,u)\}$

② $X = \{v,u\}$, $A[u] = 1$, $B[u] = \{(v,u)\}$

$POOL = \{(v,x), (v,w), (u,x), (u,w), (u,y)\}$

$(v,x) \leftrightarrow A[v] + wt(v,x) = 2$ ← min

$(v,w) \leftrightarrow A[v] + wt(v,w) = 3$

$(u,x) \leftrightarrow A[u] + wt(u,x) = 1+3 = 4$

$(u,w) \leftrightarrow A[u] + wt(u,w) = 1+4 = 5$

$(u,y) \leftrightarrow A[u] + wt(u,y) = 1+2 = 3$

$A[x] = 2$

$B[x] = B[v] \cup \{(v,x)\}$

$= \{(v,x)\}$

③ $X = \{v,u,x\}$, $A[x] = 2$, $B[x] = \{(v,x)\}$

$POOL = \{(v,w), (u,w), (u,y), (x,y)\}$

$(v,w) \leftrightarrow 3$

$(u,w) \leftrightarrow 5$

$(u,y) \leftrightarrow 3$ ← min

$(x,y) \leftrightarrow A[x] + wt(x,y) = 2+2 = 4$

$A[y] = 3$

$B[y] = B[u] + \{(u,y)\}$

$= \{(v,u), (u,y)\}$

Hence the shortest path from v to y is $\{v \to u \to y\}$ and has length of 3.

Problem 2:

a. There is no shortest path for a graph with negative weights. But if we are only allowed to traverse an edge once, then the shortest path for the graph shown is $A \rightarrow B \rightarrow C$.

b. BFS can find the shortest path in $O(n + m)$, which is faster than Dijkstra's algorithm which takes $O(mlogn)$. But to make each edge of the graph weigh 1, we must add more edges, and vertices. This total number of vertices and edges will increase by W – the weight of each edge and making the run time of BFS very slow if W is in the order of $O(n^3)$, for example.

Problem 3:

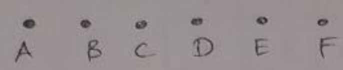
We can keep a hash table that can help with retrieving the node, based on its value. So, we can use a HashMap with a key equal to the key used in the priority queue, and a value equal to the node itself. Hence when we want to delete a node from the queue, we can find the node to be deleted using its key.

Problem 4:

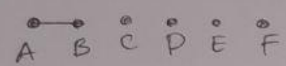
MST using Kruskal algorithm

(1) sorted edges = {(A,B), (C,D), (A,E), (B,D), (E,F), (A,F), (D,F), (B,C), (A,D)}

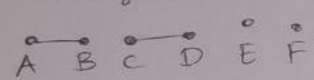
cluster:



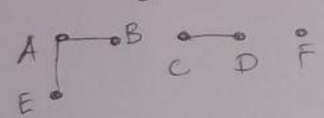
(2) using (A,B)



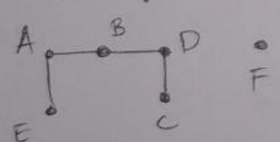
(3) using (C,D)



(4) Using (A,E)



(5) Using (B,D)



(6) Using (E,F)