Habib University - Algorithms: Design and Analysis (CS/CE 412/471) – Spring 2025

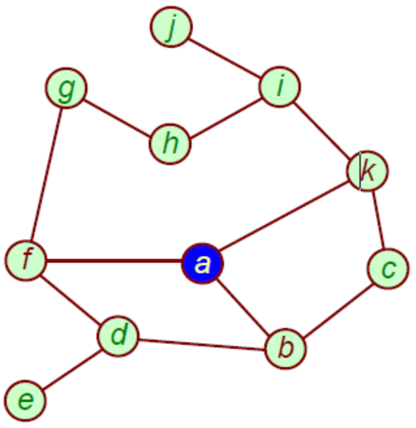
Quiz 03

Name: ____ **ID:** __________ **Section:** ________

Answer all questions on this question paper. Use back side for 1 d) and e) only.

Q1. We need to modify the **Bucket Algorithm** to perform Breadth First Search (BFS) and output a BFS Spanning Tree. A BFS Spanning Tree is a spanning tree of a graph that is constructed using Breadth-First Search (BFS). It is formed by performing a BFS traversal starting from a selected starting node and adding edges in the order they are discovered.

Answer the following:



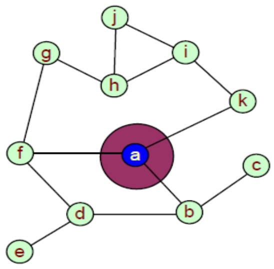
a) [1 point] State the input(s): *Adjacency matrix of an undirected Graph G, starting vertex x*

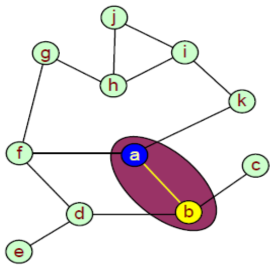
b) [1 point] State the output(s): *BFS Spanning Tree T*

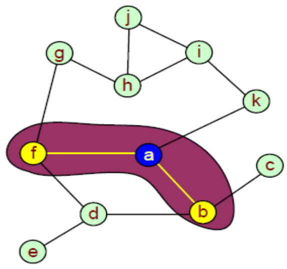
c) [5 points] Write down the modified Bucket Algorithm.

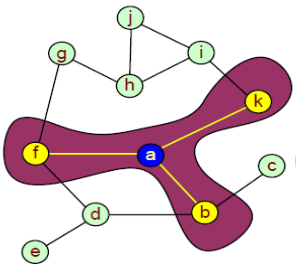
```
BUCKET-BFS(G, x):
1.  Initialize Queue Q
2.  Initialize Adjacency Matrix of a Tree T
3.  Put vertex x of Graph G in the Bucket B
4.  u = x
5.  While there are edges coming out of the Bucket B from u
        a.  Select an edge connecting vertex u in B to v not in B
                i.   Put v in B
                ii.  ENQUEUE(Q, v)
                     //Add (u, v) in T
                iii. T[u][v] = 1
                iv.  T[v][u] = 1
        b.  If no edge coming out of Bucket B from u:    // all adjacent vertices of u visited
                i.   u = DEQUEUE(Q)
6.  return T
```

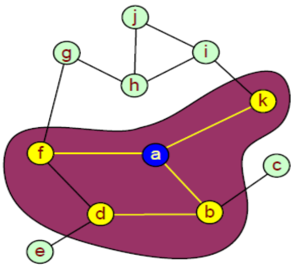
d) [5 points] Dry run your algorithm stated in part c) on the graph shown **above**, clearly showing how the spanning tree is formed. (redraw and answer on back side; show state of any additional data structure you are using). Start from vertex a. After your dry-run, highlight the edges in the BFS-ST in the diagram **above**.

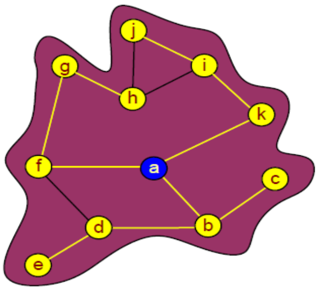

No spanning edge & One vertex in the Bucket


One spanning edge & Two vertices in the Bucket


Two spanning edges & 3 vertices in the Bucket


3 spanning edges & four vertices in the Bucket


4 spanning edges & five vertices in the Bucket


10 spanning edges & 11 vertices in the Bucket

e) [3 points] Derive the Time Complexity of your algorithm stated in part c). Justify your answer by briefly explaining.

```
1. Initialize Queue Q                                              $O(1)$
2. Initialize Adjacency Matrix of a Tree T                         $O(p^2)$
3. Put vertex x of Graph G in the Bucket B                         $O(1)$
4. u = x                                                           $O(1)$
5. While there are edges coming out of the Bucket B from u         $O(p)$
      i. Select an edge connecting vertex u in B to v not in B     $O(p)$
            a) Put v in B                                          $O(1)$
            b) ENQUEUE(Q, v)                                       $O(1)$
            //Add (u, v) in T
            c) T[u][v] = 1                                         $O(1)$
            d) T[v][u] = 1                                         $O(1)$
     ii. If no edge coming out of Bucket B from u:                 $O(p)$
            a) u = DEQUEUE(Q)                                      $O(1)$
```

Time Complexity : $O(p^2)$

Q2. a) [2 points] We need to find 2-edge shortest distances from vertex 1 to all other verices. Dry run Building Block #2 on the graph shown in Figure 1. Do not modify Figure 1, show resulting graph on Figure 2.

b) [3 points] Dry run Building Block #2 on the graph shown in Figure 2. Do not modify Figure 2, show resulting graph on Figure 3.

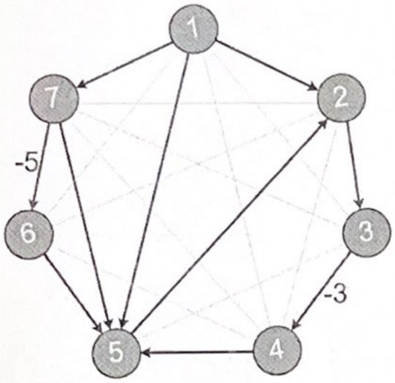


Figure 1

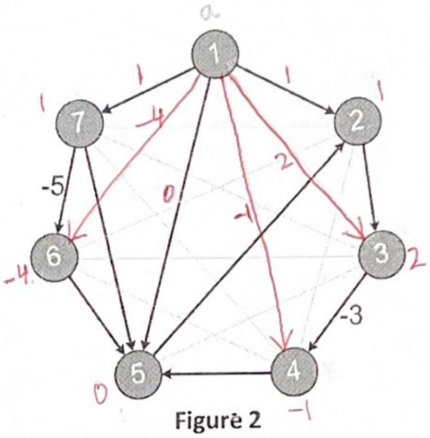


Figure 2

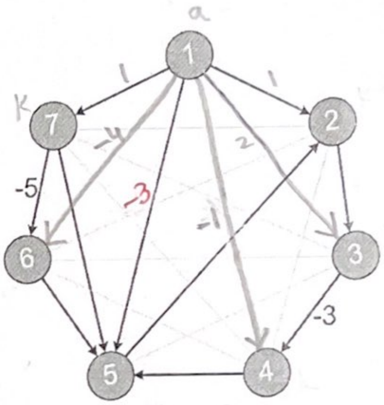


Figure 3