# COMP S265F Design and Analysis of Algorithms
# Assignment 2
# Due: May 13 (Thu), 2021, 23:59

Submit your answers in a Python program file (Question 1) and a single PDF file (Questions 2 to 4) to OLE. You can type or hand-write your answers for Questions 2 to 4.

**Question 1 (30 marks).** Given an unweighted directed graph $G = (V, E)$, two vertices $s$ and $d$, a distance $d_l \geq 0$, write a program to return the shortest path distance from vertex $s$ to vertex $d$, if the path exists and it is within distance $d_l$; return $-1$ otherwise.

You are given the program q1.py, which reads input in the following format:

- The first line contains a type letter ("u" for undirected graph / "d" for directed graph), $n$ (number of vertices), $m$ (number of edges), $s$ (source vertex), $d$ (destination vertex), distance $d_l$. These items are all separated by a space. Note that we assume the vertices are labeled by 0 to $n-1$.
- Each of the next $m$ lines contains two integer $u, v$, which indicates an edge $(u, v)$ in the graph $G$.

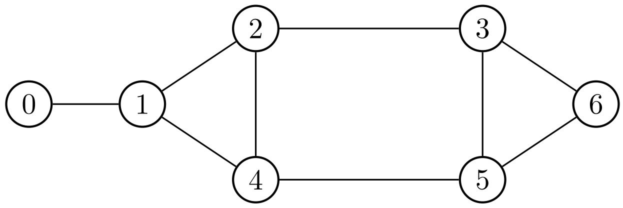
**Example 1:**

**Sample Input**

```
u 7 9 0 6 7
0 1
1 2
3 2
3 6
2 4
3 5
4 1
4 5
5 6
```

**Sample Output**

```
4
```



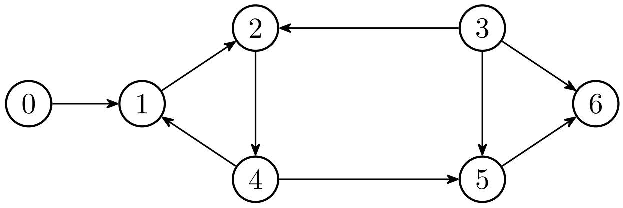
**Example 2:**

**Sample Input**

```
d 7 9 0 6 7
0 1
1 2
3 2
3 6
2 4
3 5
4 1
4 5
5 6
```

**Sample Output**

```
5
```



Complete the following functions **without changing other code in** q1.py.

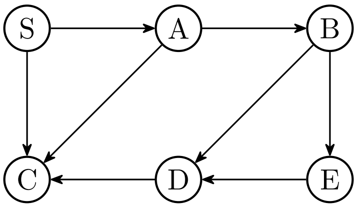
(a) The function `createGraph` creates an adjacency list `self.graph` using the edge list `self.edges`. [10]

(b) The function `bfs` performs the Breadth-First Search (BFS) and maintains the shortest path distance from $s$ to all vertices $v$ in `self.dist[`$v$`]`. [15]

(c) The function `printResult` prints the required distance output. [5]

Note that there will be multiple test cases to test your functions.

**Question 2 (15 marks).** Perform a Depth-First Search (DFS) on the following unweighted directed graph, using vertex S as the source.

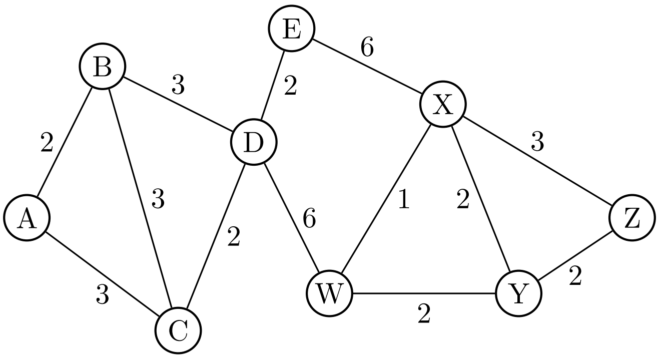


(a) List the vertices in the discovered order of DFS, and show for each vertex $v$, its discovery time $d[v]$, finish time $f[v]$, and its depth-first tree parent $\pi[v]$ in the DFS. Then, draw the depth-first tree obtained. [8]

(b) Show the classification of each edge (tree edge, back edge, forward edge, cross edge). [4]

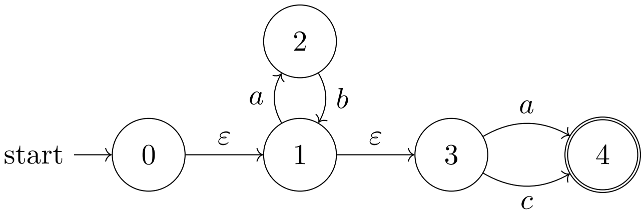
(c) Show the ordering of vertices obtained by topological sort, if any. [3]

**Question 3 (15 marks).** Use the Kruskal's algorithm to find a minimum spanning tree (MST) of the following weighted undirected graph:



Show in each step, the edge considered by the Kruskal's algorithm and whether the edge is included in the resultant MST, then draw the resultant MST, and finally compute its total weight.

**Question 4 (40 marks).** Let $\Sigma = \{a, b, c\}$ be the input alphabet of the following NFA with $\varepsilon$ moves:



(a) Write down the transition table of the above NFA including the lambda closure of the states. [10]

(b) Transform the above NFA to a DFA. Write down the transition table of the DFA. [20]

(c) Reduce the number of states of the DFA in (b), and write down the new transition table. [7]

(d) Draw the diagram of the DFA in (c). [3]