Name:

ID:

CSCI 3104, Algorithms | **Escobedo & Jahagirdar**

Homework 5A (40 points) | **Summer 2020, CU-Boulder**

Advice 1: For every problem in this class, you must justify your answer: show how you arrived at it and why it is correct. If there are assumptions you need to make along the way, state those clearly.

Advice 2: Verbal reasoning is typically insufficient for full credit. Instead, write a logical argument, in the style of a mathematical proof.

Instructions for submitting your solution:

- The solutions **should be typed**, we cannot accept hand-written solutions. Here's a short intro to **Latex**.
- In this homework we denote the asymptomatic *Big-O* notation by $\mathcal{O}$ and *Small-O* notation is represented as o.
- We recommend using online Latex editor **Overleaf**. Download the **.tex** file from Canvas and upload it on overleaf to edit.
- You should submit your work through **Gradescope** only.
- If you don't have an account on it, sign up for one using your CU email. You should have gotten an email to sign up. If your name based CU email doesn't work, try the identikey@colorado.edu version.
- Gradescope will only accept **.pdf** files (except for code files that should be submitted separately on Canvas if a problem set has them) and **try to fit your work in the box provided**.
- You cannot submit a pdf which has less pages than what we provided you as Gradescope won't allow it.

Piazza threads for hints and further discussion

Piazza Threads
Question 1
Question 2
Question 3
Question 4

Recommended reading:
Graph Algorithms Intro: Ch. 22 → 22.1, 22.2, 22.3
Graph Algorithms SSSPs: Ch. 24 → 24.3

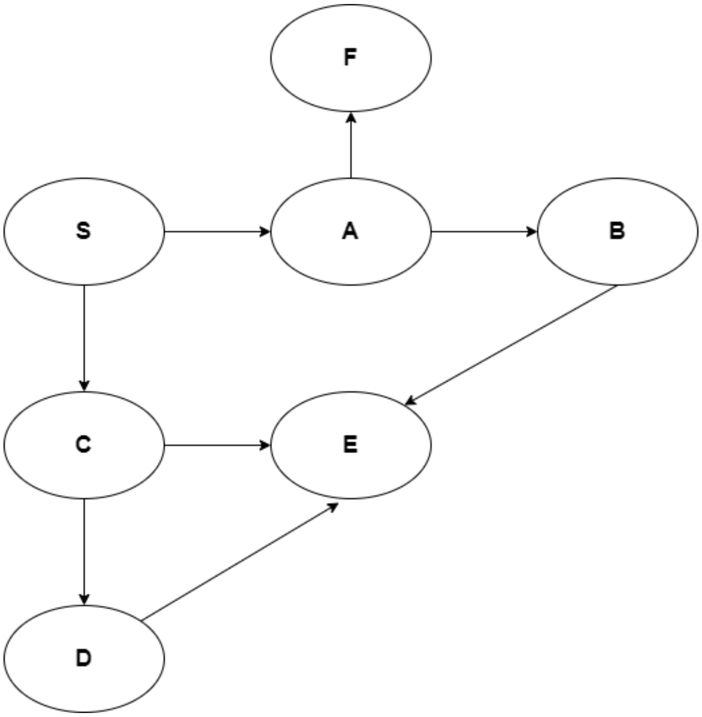
Name:

ID:

CSCI 3104, Algorithms **Escobedo & Jahagirdar**

Homework 5A (40 points) **Summer 2020, CU-Boulder**

1. (5 pts) Consider the following unweighted directed graph.



(a) (2.5 pts) Write down the order in which nodes are visited if Depth First Search (DFS) is called on the above graph with **S** as the source node.

There can be many possible DFS for the above graph. Such possible DFS are
1.S,A,F,B,E,C,D
2. S,C,D,E,A,F,B
3.S,A,B,E,F,C,D
4.S,C,D,E,A,F,B
5.S,C,E,D,A,B,F

(b) (2.5 pts) Write down the order in which nodes are visited if Breadth First Search (BFS) is called on the above graph with **S** as the source node.

Ordering of BFS are :1. S, A, C,F,B,D,E
2. S, C, A, D, E, F,B
3.S,C,A,E,D,B,F
4.S,A,C,B,F,D,E

Name:

ID:

CSCI 3104, Algorithms **Escobedo & Jahagirdar**
Homework 5A (40 points) **Summer 2020, CU-Boulder**

2. (5 pts) A simple path $s \to t$ in a graph G is a path starting at s, ending at t, and never visiting the same vertex twice. Give an example graph (simple, undirected, unweighted) $G = (V; E)$ and vertices $s, t \in V$ such that DFS finds a path from s to t which is neither a shortest path nor a longest simple path. Detail the execution of DFS (list the contents of the stack at each step, and which vertex it pops off the stack), show the final $s \to t$ path it finds, show a shorter $s \to t$ path, and a longer simple $s \to t$ path.

Assume complete graph K_4,where every vertices are having weight of 1. The nodes are numbered 1,2,3,4 and assume source is at node 1 and destination is at node 3.
Execution details of K_4: Initially stack is empty. Since 1 is the source node,it is pushed into stack.
Stack Status: 1
It has 2 choices either 2 or 4. Assume 2 is visited.**Stack Status: 1,2**
It has 2 choices either to choose node 3 or node 4. Assume 3 is visited, which is also destination. **Stack Status: 1,2,3**
Shortest path will be from 1-3,
Longest path will be either 1,2,4,3 or 1,4,2,3,
Final path : 1-2-3 or 1-4-3.

3. (20 pts) Assume you're given an integer matrix that representing islands in an ocean. A value of 0 in a particular cell indicates water and a value of 1 indicated land. An island is region of land connected vertically, horizontally, or diagonally. The size of an island is the total number of connected land cells. Write an algorithm to compute the sizes of all islands in the matrix. Example:

```
1 0 0 1
0 1 0 1
0 1 0 0
0 1 0 1
```

would output 1, 2, 4.

(a) (6 pts) State if the graph data structure that your algorithm will use for this problem will be i) be weighted or unweighted and ii) directed or undirected. In addition, iii) what makes two nodes have an edge/connection? Give an explanation for each of your answers.

The graph will be (i)unweighted, as the adjacency matrix doesn't have values other than 0 or 1.
(ii) directed, as the required output is one of the possibility.It is possible, only when the graph is directed. Assume I_i, $\forall i \in 1, 2, 3, 4$ is represented by Nodes 1,2,3,4. The nodes 1 have outgoing edge, 2 and 4 has bidirectional edges.
(iii) 1 in the adjacency matrix represents the island and edges between the island and the both are connected.0 represents the water and it is not connected to any other island

(b) (4 pts) Provide a 3-4 sentence description of how your algorithm works, including how the matrix is converted to the graph, how adjacent vertices are identified, and how the algorithm traverses the graph to identify connected vertices.

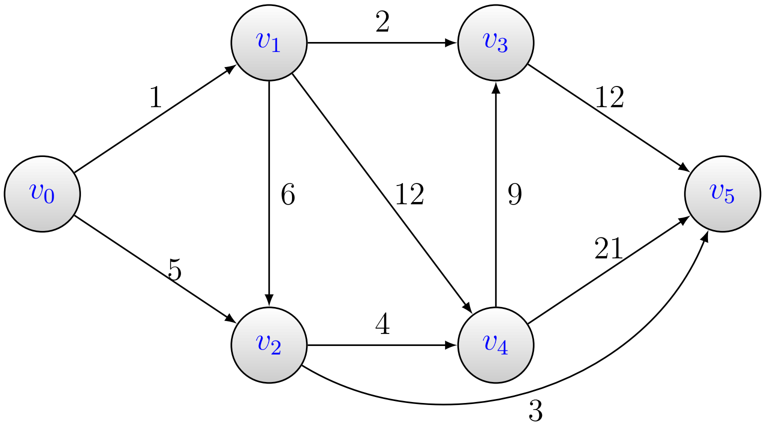
The 1's in the adjacency matrix represents island,which are connected with another island and O's represents the island,which are not connnected. Assume I_i, $\forall i \in 1, 2, 3, 4$ is represented by Nodes 1,2,3,4. The node 1 have outgoing edge, node 2 and 4 has bidirectional edges and node 3 is not connected to any other island. Node 3 will be one component and Node 1,2,4 will be another component. The algorithm uses Depth-First Search (DFS) on the connected components.

(c) (10 pts) Write well commented pseudo or real code to solve this problem.

Pseudo code will consists of 3 procedure(or function). Procedure DFS will be used to find the order of the connected components of adjacent neighbours. Procedure Visited : To check if the given row,col must be included in the DFS and not yet visited. Procedure print: The order of connected components.

```
$ROW \leftarrow 4$
$COL \leftarrow 4$
function VISITED(M[][COL],r,bool visited[][COL])    ▷ row number is in range, column number is in range and value is 1
and not yet visited
    return $(row \geq 0) and (row < ROW) and (col \geq 0) and (col < COL)$
end function
function DFS(M[][COL],r,c,bool visited[][COL])    ▷ Function of DFS,which will consider 8 adjacent neighbours
    Array $rowNbr = \{-1, -1, -1, 0, 0, 1, 1, 1\}$    ▷ These arrays are used to get row and column numbers of 8 neighbours of a given cell
    Array $colNbr = \{-1, 0, 1, -1, 1, -1, 0, 1\}$
    Array $visited[r][c] \leftarrow true$    ▷ Mark this cell as visited

    for $k \leftarrow 0$ to $k < 8$ do

        if VISITED(M, r + rowNbr[k], c+ colNbr[k], visited) then
      DFS(M, r + rowNbr[k], c + colNbr[k], visited)
        end if
    end for
end function
function PRINT(M[][COL]))    ▷ To print the order of island
    $visited[ROW][COL]$

    for $i \leftarrow 0$ to $ROW$ do

        for $j \leftarrow 0$ to $COL$ do

            if M[i][j] and visited[i][j]==False then DFS(M, i, j, visited)
            end if
print Order
        end for
    end for
    end function
```

4. (10 pts) Using Dijkstra's algorithm, determine the length of the shortest path from v_0 to each of the other vertices in the graph. Clearly specify the distances from v_0 to each vertex **after each iteration** of the algorithm.



Shortest path from v_0 to every node is represented in the form of table

v_1	v_2	v_3	v_4	v_5
1	5	3	9	8

Initially all the vertices are having distance ∞. Source is initialized, which will have distance 0.
1^{st}iteration, two choices so each adjacent vertices distance are updated as $min(1, \infty)$ for v_1 and $min(5, \infty)$ for v_2.Remaining vertices are ∞.
2^{nd}iteration, 3 path from v_1 are updated as $min(2, \infty)$ for v_3, $min(5, 7)$ for v_2, $min(13, \infty)$ for v_4. v_5 still remains ∞
3^{rd}iteration, 2 path from v_2 are updated as $min(9, 13)$ for v_4, min(8,∞)v$_5$.
4^{th}iteration, 2 path from v_4 are updated as $min(3, 18)$ for v_3, $min(8, 30)$ v_5.

5. ***Extra Credit (5% of total homework grade)*** *For this extra credit question, please refer the leetcode link provided below or click here. Multiple solutions exist to this question ranging from brute force to the most optimal one. Points will be provided based on Time and Space Complexities relative to that of the most optimal solution.*

   Please provide your solution with proper comments which carries points as well.

   https://leetcode.com/problems/keys-and-rooms/

   ```
   Replace this text with your source code inside of the .tex document
   ```