American University of Sharjah
School of Engineering
Department of Computer Engineering
P. O. Box 26666
Sharjah, UAE

Instructor: Omar Arif
Office: ESB-2178
Phone: 971-6-515 4821
e-mail: oarif@aus.edu
Semester: Fall 2024

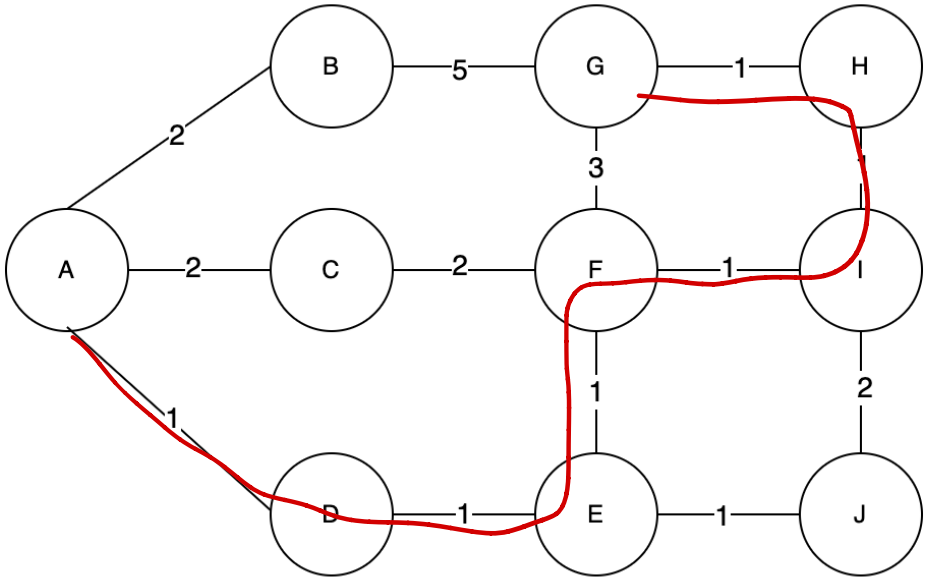
Name: _____

ID : _____

CMP333 – Quiz 1
Time: 30 minutes
Total Points: 13

Question 1 (10 points)

Consider the graph below, where **A** is the start state and **G** is the goal state. Lines between nodes are labeled with the cost to traverse them.
For each search strategy {BFS, DFS, Uniform Cost}, list in order the states pushed onto the explored list and the path found.
(1) In all search problems, use **alphabetical order** to add nodes to the frontier.
(2) Use Graph-Search for all search strategies



_____

Breadth first Search [5 points]

Order of states pushed onto the explored list __ABCDG___ ____

Path found: _ABG_____

Depth First Search [5 points]

Order of states pushed onto the explored list _____ADEJIHG_____

BFS: FIFO

Frontier: ~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ F G

Explored: A B C D G

Path found: A → B → G.

DFS: LIFO.

Frontier: ~~A~~ B C ~~D~~ ~~E~~ F ~~J~~ ~~I~~ F ~~H~~ ~~G~~

Explored: A D E J I H G

Path: A → D → E → J → I → H → G

UCS:

Frontier: ~~A~~ ~~B~~ ~~C~~ ~~D~~ ~~E~~ G ~~F~~ ~~F~~ ~~J~~ G ~~I~~ ~~I~~ ~~H~~ G

1 1 1 1+1 2+5 2+2 1+1+1 1+1+1 1+1+1+3 1+1+1+1 1+1+1+2 1+1+1+1+1

2 7 4 3 3 6 4 5 5 6

Same cost.

go for less nodes.

Explored: A D B C E F J I H G

Path: A D E F G

Path found: _____ADEJIHG_

Uniform Cost Search [5 points]

Order of states pushed onto the explored list ____ADBCEFJIHG___

Path found: _____ADEFG

Question 2 (3 points)
Assume you are given **m x n** 2D binary grid, which represents a map of **1's** (land) and **0's** (water).
An island is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Also assume you have access to a function function `bfs(grid, i, j)` defined below

Example 1:

```
Input: grid = [
  ["1","1","1","1","0"],
  ["1","1","0","1","0"],
  ["1","1","0","0","0"],
  ["0","0","0","0","0"]
]
Output: 1
```

Example 2:

```
Input: grid = [
  ["1","1","0","0","0"],
  ["1","1","0","0","0"],
  ["0","0","1","0","0"],
  ["0","0","0","1","1"]
]
Output: 3
```

```
def bfs(grid, i, j):
    frontier = [(i, j)]
    while frontier is not empty:
      x, y = frontier.pop(0)
      for dx, dy in [(1, 0), (-1, 0), (0, 1), (0, -1)]:
        nx, ny = x + dx, y + dy
        if 0 <= nx < len(grid) and 0 <= ny < len(grid[0]) and
                                    grid[nx][ny] == 1:
          grid[nx][ny] = 0
          frontier.append((nx, ny))
```

How will you use this function to find the number of islands in the grid? Write pseudocode.

```
count = 0
for i in range(len(grid)):
        for j in range(len(grid[0])):
            if grid[i][j] == 1:
                bfs(grid, i, j)

print("No of islands: ", count)
```

Just ignore this I guess.