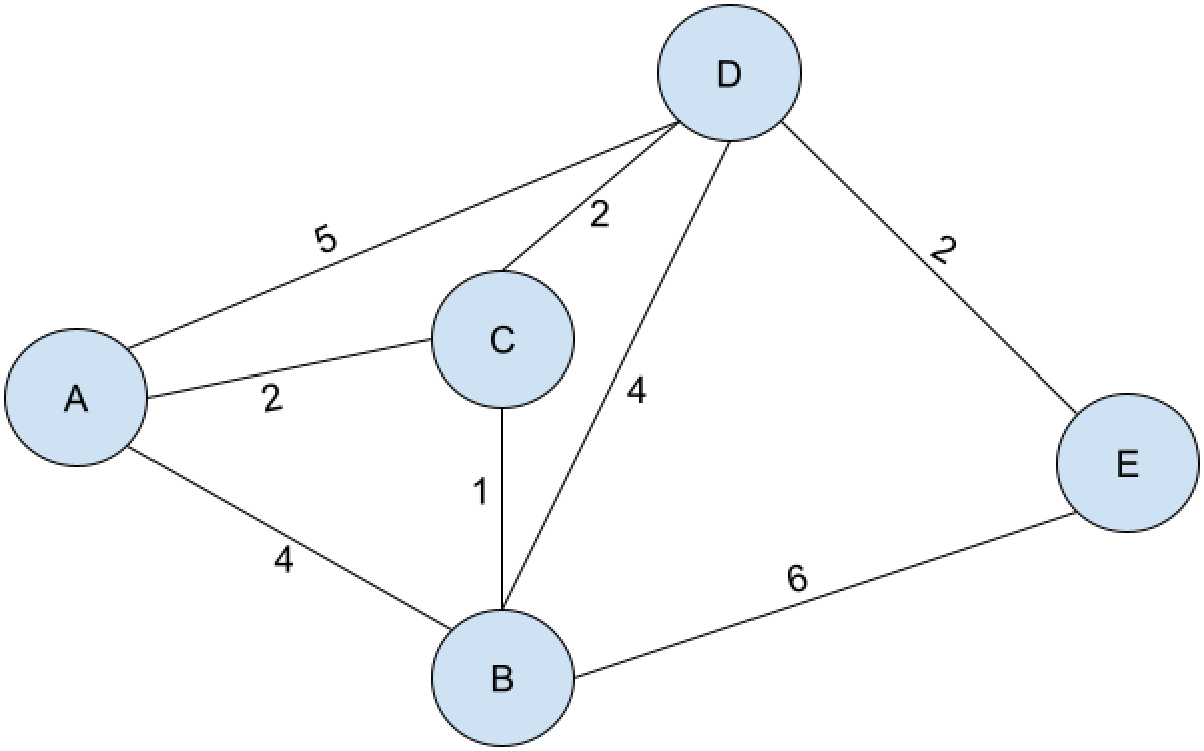
Question - 1 ***[8 Points]***



Given an undirected graph with 5 Nodes: A, B, C, D, and E. The numbers on the edge of the graph represent the edge weight. Considering the **Source vertex as B** and the **Destination vertex as D**, perform **Uniform Cost Search** (with some additional rules, see below) in order to fill in 3 data structures, namely FRONTIER_LIST, EXPLORED_LIST, and POP_LIST. The description of these data structures is as follows:

- FRONTIER_LIST - similar to the “frontier” in Canvas video lectures with **one key difference -** once an entry is made in the FRONTIER_LIST, it will NOT be removed later as the algorithm progresses. Thus, when you add a node in “frontier”, make an entry corresponding to that node in the FRONTIER_LIST. But when you pop a node from “frontier”, DO NOT try to remove any entry from the FRONTIER_LIST
- EXPLORED_LIST - this data structure has the same interpretation of “explored list” as in the Canvas video lecture. Thus, when a node is visited, make an entry to this data structure
- POP_LIST - after you pop a node from the “frontier”, make an entry in the POP_LIST **immediately** to denote the popped node

You will track these 3 data structures in addition to the standard data structures used by UCS in the Canvas video lecture.

The **entries** that go in these data structures are the Nodes appended with the sequence number, where the sequence number is the order in which you will enter that particular node in one of these data structures.

For example, in a hypothetical scenario, if the first 2 steps of your algorithm require adding Node A to the FRONTIER_LIST and adding Node B to the POP_LIST respectively, you will make A1 as an entry in FRONTIER_LIST and B2 as an entry in the POP_LIST. If in the next step, you are adding B to the EXPLORED_LIST, an entry B3 will be made in the EXPLORED_LIST

The values will be filled in these data structures until the UCS search stops (i.e., the last value will be corresponding to the destination node being popped from the frontier).

There are some additional rules for performing UCS that you need to follow filling these 3 data structures:

1. In the UCS algorithm, while performing any operation like adding a node to the frontier or popping a node from the frontier, etc., to decide the order in which to consider the nodes, always select the node with a lower edge weight before the node with a higher edge weight and in case of a tie among edge weights, break the tie by selecting the nodes in alphabetical order
2. When exploring a node, first pop it from the "frontier," add it to the "popped list", and then add it to the "explored list"
3. When updating the cost associated with a node in FRONTIER_LIST with a lower cost, simply replace the old cost with the new cost. Do not add the node to FRONTIER_LIST again

***[6 Points]*** **1.1** Now, to help you get started with answering the question, here are some values that are part of the correct answer and go into each of the data structures. Complete these lists and finally answer the question in the Solution Matrix

- FRONTIER_LIST: B1, C4,.........
- EXPLORED_LIST: B3,.........
- POP_LIST: B2,........

You don't need to show your entries in these data structures and you won't be graded for the entries made in the lists. You will **ONLY be graded for what you will fill in the Solution Matrix**.

The rows of the Solution Matrix denote Nodes and columns denote Sequence Numbers. Fill in the values **F, EX,** or **P** in the answer cells. Fill the cells of the Solution Matrix corresponding to the correct answers. Here, FRONTIER_LIST contains B1 and C4, therefore **F** is filled in the cell corresponding to Node B and Sequence number 1 and in the cell corresponding to Node C and Sequence Number 4. B3 is in EXPLORED_LIST, so **EX** is filled corresponding to Node B and sequence number 3. B2 is in POP_LIST, so **P** is filled corresponding to Node B and sequence number 2.

Please **don't fill anything in the cells which is not a part of your answer**. Filling anything other than the permissible values (F, EX, and P) will lead to a **0 score** in this part.

**Solution Matrix**

| Sequence Numbers | Node A | Node B | Node C | Node D | Node E |
|---|---|---|---|---|---|
| 1 | | F | | | |
| 2 | | P | | | |
| 3 | | EX | | | |
| 4 | | | F | | |
| 5 | F | | | | |
| 6 | | | | F | |
| 7 | | | | | F |
| 8 | | | P | | |
| 9 | | | EX | | |
| 10 | P | | | | |
| 11 | EX | | | | |
| 12 | | | | P | |
| 13 | | | | (EX) | |
| 14 | | | | | |

Nodes are added to any of the 3 lists: FRONTIER_LIST, EXPLORE_LIST or POP_LIST one at a time, and thus entries made in these lists have different sequence numbers.

Points are awarded if your entries match the answers.

- full credits are awarded in both the cases: when the cell corresponding to Node D Sequence Number 13 is left blank and also when it is filled with EX
- Full points are awarded when 8 (9) correct entries are made from rows 5 to row 12 (13) and the last 2 (1) rows are left blank.
- 0.4 points for each of your answers matching with the entries/blanks in the solution
- It was expected to leave the last 2 rows blank

***[0.5 Points]*** **1.2** When the search ends, the cost of reaching the following nodes is:

- Node A is .........3..........
- Node D is .........3..........

0.25 points for each correct entry filled. No partial marks if your answers don't match

***[1.5 Point]*** **1.3** Now if the Source Node is D, Destination is E, and the goal is to perform **Depth First Search**. Write the sequence of nodes that will be explored. For this part, ignore the edge weights and select the nodes in alphabetical order. Don't repeat the nodes which are once visited

.........D, A, B, C, E.................................................................

- Full points are also awarded to answers missing either source and destination nodes, i.e. to A, B, C, E and D, A, B, C
- Partial points are awarded for each node when its order matches with the solution, starting from the source node (D)

Question - 2 ***[2 Points]***

Given below is a 4x4 grid. Perform A* search and find the **lowest cost path** from the source to reach the destination.

Start with A (source) and reach P (destination). In this 4x4 grid shown below, you are allowed to move only 1 step to reach an adjacent cell (sideways, up or down, and diagonal) at a time. The cost of moving to an adjacent cell is 1 unit. The coordinates of A are (0,0) while the coordinates of P are (3,3). Similarly, the coordinates of B is (0,1) whereas that of E is (1,0)

The cells colored yellow are different. If you move to any of these cells, the cost is 4 units

| | | | |
|---|---|---|---|
| A | B | C | D |
| E | F | G | H |
| I | J | K | L |
| M | N | O | P |

However, upon reaching any **intermediate** cell, you can fly directly to your destination as you want to minimize the total cost of reaching the destination. While flying, you should follow a straight-line path to the destination i.e. use the Euclidean distance heuristic. The cost of flying is equal to the Euclidean distance from that node to the destination.

Any cell where you reach after any sequence of 1-step moves is an intermediate cell. You can fly from the source cell itself but you cannot fly from the destination cell. You are NOT allowed to fly from one intermediate cell to another intermediate cell, i.e. at any point when you decide to fly, you will fly towards the goal.

Find the lowest cost route to reach P from A. Fill in the blank by writing the names of cells in the order in which you will reach them separated by commas (,) For e.g. - if you move from A to E and then fly from E to P, your answer would be: A, E, P

…A, B, G, P………………………………………………………………….

The question was later changed to find only the lowest cost route.

Full credits are awarded even if you missed this clarification and performed A* search. In this case, the correct answer was A, P

- No partial points in this part