Assignment 1: Informed and Uninformed Search
Due Date: 11.59 PM 09/30/2022

1 Instructions

For this assignment, we are going to look at informed and uninformed search algorithms. For the first question ('Graph Traversals'), you do not need to submit code. For the second question ('Water Jug Problem'), you need to provide both code and answers to the questions asked.

1.1 What to submit?

A *.zip* file that contains a text file (*.docx* or *.pdf*) with all the answers to the questions, your code package, a text file (*.txt* or other) with the output of the coding questions, any other artifacts that you think would be helpful for evaluation.

2 Graph Traversals (25 points)

For this question, please refer to Figure 1. The value for h on each node represents the output of an unknown heuristic function for each node. The values on top of each edge is the weight for that particular edge. For example, the weight for edge BC is 5, and the weight for edge AC is 4. S is the start node of the graph, and G is the goal node. Using all the information presented in the graph, answer the following questions.

1. What is the branching factor for this graph? (2)
2. Based on the heuristic values, is the heuristic function:
   - Admissible? If not, why? (4)
   - Consistent? If not, why? (4)
3. What is the minimum cost of moving from S to the goal G? Provide an example of one of the paths that has the minimum cost. (3)
4. What is the minimum length of all path(s) from S to the goal G? Provide an example of one of the paths that has the minimum length. (3)
5. What is the sequence of nodes in the stack/frontier when you apply Iterative Deepening on this graph to find a path from S to G? Is a path is found, what is the cost of the path? How does it compare with the optimum cost? (No code required) (9)

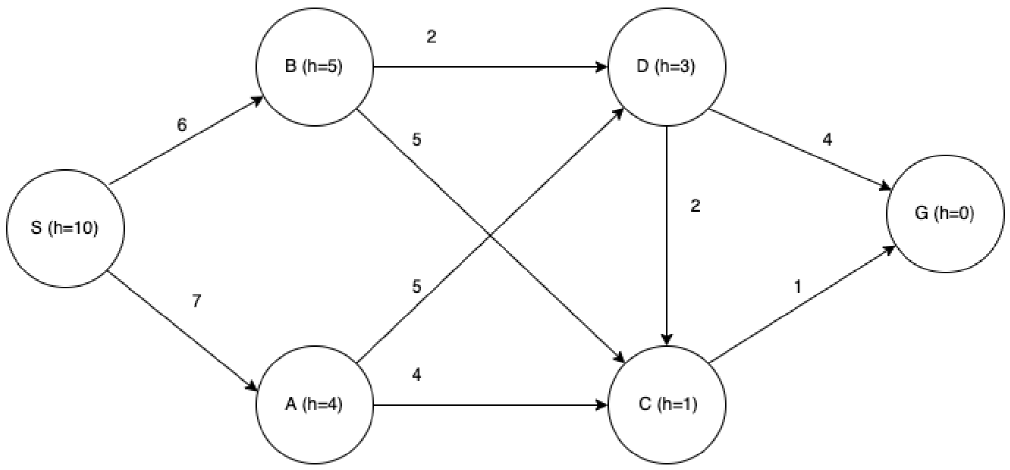

Figure 1:

3 Water Jug Problem (75 points)

3.1 Problem Statement

There are two jugs J1, J2 with Capacity 5 gal and 2 gal respectively.
Initial State: J1 has 4 gallons of water and J2 has 1 gallon of water.
Actions:

1. **Pour1** Pour from jug X to jug Y until X empty or Y full. (if Y is already full/ X is empty, do not take this action)
2. **Pour2** Pour from jug Y to jug X until X is full or Y is empty.(if X is already full/Y is empty, do not take this action)
3. **Dump1** Empty jug X
4. **Dump2** Empty jug Y

Goal States: The goal states that need to be reached are

1. (0,1) (Jug 1 has 0, Jug 2 has 1)
2. (4,0) (Jug 1 has 4, Jug 2 has 0)
3. (5,0) (Jug 1 has 5, Jug 2 has 0)
4. (3,2) (Jug 1 has 3, Jug 2 has 2)
5. (1,2) (Jug 1 has 1, Jug 2 has 2)

3.2 Coding Assignment

For the problem described above, you need to provide the code and the output for the algorithms that you implement. Python 3+ is recommended. Please provide detailed instructions on how to run your code. In case you are not familiar with python, please reach out to me or the TA and provide us with additional instructions on how to run your code.

The coding assignment carries a total of **50 points**.

Given each of the initial to goal state pair, you need to find a solution path using the following algorithms:

1. Breadth First Search(BFS)
2. Depth First Search(DFS) (without tracking seen nodes)
3. A^* Algorithm

You **can not** use zero-heuristic to implement A*.

3.2.1 Output

Your code should output a ”output.txt” file.

For each of the initial to goal state pair, the output.txt file should contain details about the path, sequence of actions and cost of solution path for each of the 3 algorithms. The **cost** is 1 for each action. For example, if two actions are required to reach the goal from the source the cost of this path will be 2.

Let the initial state be (1, 1) and the goal state be (0, 0). One possible path from initial to goal state would be $(1, 1) - -Dump1 - - > (0, 1) - -Dump2 - - > (0, 0)$

In this case, your code should output a **”output.txt”** file of the following format for each initial-goal state pair and search strategy:

```
Initial State : (1, 1)
Goal state: (0, 0)

Searching strategy: BFS, DFS or A*
Path: (1, 1), (0,1), (0, 0)
Action: Dump1, Dump2
Cost: 2
```

3.2.2 Notes

1. You need to represent the problem space in a graph space. Think about how to achieve it. You need a well-defined initial state, goal state and sequence of actions. Actions are only legal under certain conditions. Remember to enforce these conditions. Follow the slides/discussions from our class
2. As you might remember from class, that some algorithms are not complete i.e. they may not find a solution. I suggest you write a timeout function to deal with this.

3.3 Questions

Along with your coding assignment, please provide an answer to the following questions:

1. Describe the heuristic function used in your A* search.
2. Is the heuristic you used admissible? Describe why.
3. For each of the 5 initial to goal state pair, did BFS, DFS and/or A* reach a solution? If not, describe why?
4. For each of the 5 initial to goal state pair, is the BFS, DFS and/or A* solution optimal? Describe why.
5. How would you convert the A* search algorithm into Uniform Cost Search?

Each question carries **5 points**.