AI ASSIGNMENT 2
Date of Submission : 19/09/2023

Theory

NOTE: Marks are reduced for Unclear or Unsound explanation

1. Suppose you decide not to keep the explored set (i.e. the place where the explored nodes are saved) in the A* algorithm.

   1. If the heuristic used is admissible, will the new algorithm still be guaranteed to find the optimal solution? Explain why (or why not). Suppose you decide not to keep the explored set (i.e. the place where the explored nodes are saved) in the A* algorithm.

   2. Will the new algorithm be complete? Explain.

   3. Will the new algorithm be faster or not? Explain

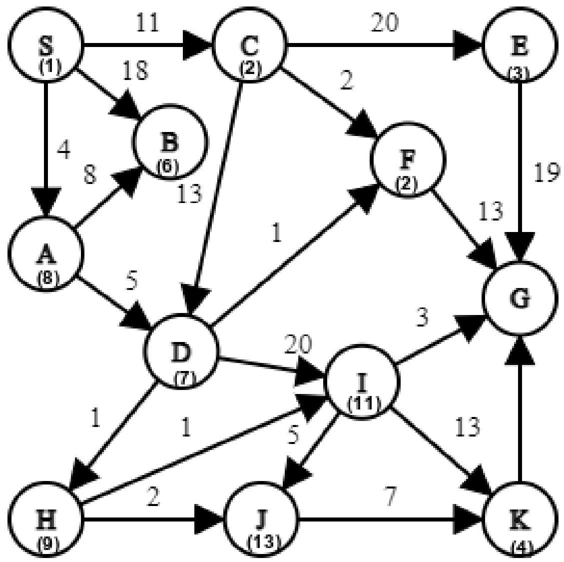
   Answer:

   Ans1: if you remove the explored set from A* while keeping an admissible heuristic, you can't guarantee that the modified algorithm will always find the optimal solution. It may still find the optimal solution in some cases, but it could also perform much worse in terms of time and space complexity due to revisiting nodes and losing track of optimal paths.

   Ans2: The new algorithm will not be complete because without the explored set, it may revisit previously explored nodes, leading to infinite loops in certain cases.

   Ans3: The new algorithm might be faster due to reduced memory overhead, but it may explore more nodes, making it less efficient in terms of time complexity compared to A*.

2. Given the graph, find the cost-effective path from S to G. S is the source node, and G is the goal node using A*, Bfs, and Dijkstra algorithm. Heuristic for Goal Node G is 0. Distance from K to G is 6.



Answer Best-First Search
S is source node so write all successors of S
A: 8, B:6, C:2
Expand which is having low heuristic
C is min, expand C
E: 3, D:7, F:2
F is min, expand F

Therefore reached the goal node, hence S- C -F-G is final path and **Total distance need to travel from S-G is 11+2+13=26**

Answer A* Algorithm
S is the source node, so write all successors of S
S-A: 8+4=12, S-B: 6+18=24, S-C: 2+11=13
Expand, which is having low f(n)
Expand S-A, i.e., Node A
S-A-D: 4+5+7=16, S-A-B: 4+8+6=18
Expand S-C, i.e., Node C
S-C-F: 11+2+2, S-C-E: 11+20+3=34, S-C-D=11+13+7=31
Expand S-C-F, i.e., Node F
S-C-F-G: 11+2+13=26
Here, G is the Goal Node, but we will continue till G is not explored.
Expand S-A-D i.e Node D
S-A-D-F: 4+5+1+2=12, S-A-D-I: 4+5+20+11=40, S-A-D-H: 4+5+1+9=19
Expand S-A-D-F i.e Node F
S-A-D-F-G: 23
Expand S-A-B i.e. Node B

No Path
Expand S-A-D-H i.e Node H
S-A-D-H-J: 4+5+1+2+13=25, S-A-D-H-I: 4+5+1+1+11=22
Expand S-A-D-H-I i.e Node I
S-A-D-H-I-J: 4+5+1+1+5+13=29, S-A-D-H-I-G: 4+5+1+1+3=14, S-A-D-H-I-K: 4+5+1+1+13+4=2
Expand S-A-D-H-I-G i.e. Node G
Since G is the Goal state, we have found the shortest distance from Source S to the target G as **S-A-D-H-I-G with a value of 14.**

Dijikstra's Algorithm

S-A : 4
S-B :12
S-C : 11
S-E : 31
S-F : 10
S-G: 14
S-D: 9
S-I: 11
S-J: 12
S-K : 19
S-H :10
Shortest path from S-G is S-A-D-H-I-G
The length of shortest path from S-G is 14.

Computational

NOTE : There are Marks for VIVA in the computational part

1.