########################

# Problem Set 2 | Search

# Group Submission

# Members:

# Yasser Parambathkandy - (G01294910)

# Indranil Pal - (G01235186)

# Date: 09/20/2022

Use the graph given in figure 1 for the next few questions. Nodes are labeled with their names, edges with

their weights. The starting state is A" and the goal state is F". When describing actions, use the notation

A->C" to refer to moving from state A to state C", which in this graph has a cost of 6. Use alphabetical

order in the case of ties.

1. Find a path from the start (A) to the goal (F) using the Depth First Search version of the Generic

Search Algorithm discussed in class, making sure to

(a) Show the status of the open and closed data structures for each iteration.

(b) Show the final search tree after the goal state is reached.

(c) Show the final result (path).

Diagram

Description automatically generated

(a)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Level 0 | A | A | A | A | A | A | A |
| Level 1 | E | B | D | D | C | C | C |
| Level 2 | F | E | B | E | E | D | B |
| Level 3 |  | F | E | F | F | B | E |
| Level 4 |  |  | F |  |  | E | F |
| Level 5 |  |  |  |  |  | F |  |

|  |  |  |
| --- | --- | --- |
| STATE | Open | Closed |
| Start | A | [] |
| A->E | E | A |
| E->F | F | A, E |
| A->B | B | A, E, F |
| B->E | E | A, E, F, B |
| E->F | F | A, E, F, B |
| A->D | D | A, E, F, B |
| D->B | B | A, E, F, B, D |
| B->E | E | A, E, F, B, D |
| E->F | F | A, E, F, B, D |
| D->E | E | A, E, F, B, D |
| E->F | F | A, E, F, B, D |
| A->C | C | A, E, F, B, D |
| C->D | D | A, E, F, B, D, C |
| D->B | B | A, E, F, B, D, C |
| B->E | E | A, E, F, B, D, C |
| E->F | F | A, E, F, B, D, C |
| C->B | B | A, E, F, B, D, C |
| B->E | E | A, E, F, B, D, C |
| E->F | F | A, E, F, B, D, C |
| C->E | E | A, E, F, B, D, C |
| E->F | F | A, E, F, B, D, C |

b)

Diagram, schematic

Description automatically generated

c) A->E->F->B->D->C

2. Answer the same three questions for the Uniform Cost Search version.

a) Show the status of the open and closed data structures for each iteration.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STATE | Options | PATH | OPEN | CLOSED | COST (g(n)) |
| Start | - | A |  |  | 0 |
| A->C | A->E cost 11  A->B cost 9  A->D cost 6  A->C Cost 1 | A->C | C | A | 1 |
| C->D | C->D cost 1  C->B cost 4  C->E cost 6 | A->C->D | D | A, C | 1+1=2 |
| D->B | D->B cost 1  D->E cost 3 | A->C->D->B | B | A, C, D | 1+1+1=3 |
| B->E | - | A->C->D ->B->E | E | A, C, D, B | 1+1+1+1=4 |
| E->F | - | A->C->D ->B->E->F | F | A, C, D, B, E | 1+1+1+1+19=23 |
| Goal | - | - | - | A, C, D, B, E, F |  |

b)

Diagram

Description automatically generated

c) A->C->D ->B->E->F

3. Answer the same three questions for A\* search using the following heuristic:

State h(State)

|  |  |
| --- | --- |
| State | h(State) |
| A | 23 |
| B | 3 |
| C | 13 |
| D | 7 |
| E | 0 |
| F | 0 |

Start

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| node | G(n) | h(n) | F(n) | prev |
| A | 0 | 23 | 23 | None |
| B | 9 | 3 | 12 | A |
| C | 1 | 13 | 14 | A |
| D | 6 | 7 | 13 | A |
| E | 11 | 0 | 11 | A |
| F | 19 | 0 |  | None |

Visited list

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| node | Path | G(n) | h(n) | F(n) | | prev |
| A | A | 0 | 23 | 23 | None | |
| E | A->E | 11 | 0 | 11 | A | |

Unvisited list

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| node | G(n) | h(n) | F(n) | prev |
| B | 9 | 3 | 12 | A |
| C | 1 | 13 | 14 | A |
| D | 6 | 7 | 13 | A |
| F | 19 | 0 |  | None |

At point E, only option is to go to F

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| node | Path | G(n) | h(n) | F(n) | | prev |
| A | A | 0 | 23 | 23 | None | |
| E | A->E | 0+11=11 | 0 | 11 | A | |
| F | A->E->F | 0+11+19=30 | 0 | 30 | E | |

Alternate 1 :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| node | Path | G(n) | h(n) | F(n) | | prev |
| A | A | 0 | 23 | 23 | None | |
| B | A->B | 9 | 3 | 12 | A | |
| E | A->B->E | 9+1=10 | 0 | 10 | B | |
| F | A->B->E->F | 9+1+19=29 | 0 | 29 | E | |

Alternate 2 :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| node | Path | G(n) | h(n) | F(n) | | prev |
| A | A | 0 | 23 | 23 | None | |
| D | A->D | 6 | 7 | 13 | A | |
| E | A->D->E | 6+3=9 | 0 | 9 | D | |
| B | A->D->B | 6+1=7 | 3 | 10 | D | |
| F | A->D->E->F | 6+3+19=28 | 0 | 28 | E | |
| E | A->D->B->E | 6+1+1=8 | 0 | 8 | B | |
| F | A->D->B->E->F | 6+1+1+19=27 | 0 | 27 | E | |

Alternate 3 :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| node | Path | G(n) | h(n) | F(n) | | prev |
| A | A | 0 | 23 | 23 | None | |
| C | A->C | 1 | 13 | 14 | A | |
| E | A->C->E | 1+6=7 | 0 | 7 | C | |
| B | A->C->B | 1+4=5 | 3 | 8 | C | |
| D | A->C->D | 1+1=2 | 7 | 9 | C | |
| F | A->C->E->F | 1+6+19=26 | 0 | 26 | E | |
| E | A->C->B->E | 1+4+1=6 | 0 | 6 | B | |
| D | A->C->D->E | 1+1+3=5 | 0 | 5 | D | |
| B | A->C->D->B | 1+1+1=3 | 3 | 6 | D | |
| F | A->C->B->E->F | 1+4+1+19=25 | 0 | 25 | E | |
| E | A->C->D->B->E | 1+1+1+1=4 | 0 | 4 | B | |
| F | A->C->D->B->E->F | 1+1+1+1+19=23 | 0 | 23 | E | |

***There are shorter routes available A->C->D->B->E->F, should we take that ?***

Diagram

Description automatically generated

4. Is this heuristic admissible? Explain why or why not.

|  |  |  |  |
| --- | --- | --- | --- |
| Path | Cost(h\*(n) | H(n) | Admissible ? (H(n) < H\*(n) |
| A->F | 30/23 (in 2 different path) | 23 | Yes |
| B->F | 20 | 3 | Yes |
| C->F (C->B->D->E->F) | 22 | 13 | Yes |
| D->F(D->B-E-F) | 21 | 7 | Yes |
| E-F | 19 | 0 | Yes |

Its admissible.

5. This graph is part of a family of similar graphs known as Martelli's family that is specifically designed

to illustrate why consistency is important for a heuristic. Replace the heuristic function from question

3 with one that is consistent and re-do the search. How was the search different?

Consistent if h(n) ≤ c(n,a,n') + h(n')

|  |  |
| --- | --- |
| State | h(State) |
| A | 23 |
| B | 3 |
| C | 13 |
| D | 7 |
| E | 0 |
| F | 0 |

Diagram

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Path | h(n) | C(n,a,n’) | h{n’) | C(n,a,n’) + h(n’) | h(n) ≤ c(n,a,n') + h(n') |
| A->B | 23 | 9 | 3 | 12 | No |
| A->C | 23 | 1 | 13 | 14 | No |
| A->D | 23 | 6 | 7 | 13 | No |
| A->E | 23 | 11 | 0 | 11 | No |
| C->D | 13 | 1 | 7 | 8 | No |
| C->B | 13 | 4 | 3 | 7 | No |
| B->E | 3 | 1 | 0 | 1 | No |
| C->E | 13 | 6 | 0 | 6 | No |
| D->B | 7 | 1 | 3 | 4 | No |
| D->E | 7 | 1 | 0 | 1 | No |
| E->F | 0 | 19 | 0 | 19 | Yes |

So, its not consistent

6. Our friendly vacuum robot needs to clean the room illustrated in Figure 2. For the purposes of this

example, the vacuum is running the entire time, so the only actions are to move between adjacent cells.

The robot's goal is to visit each of the spaces with dirt, and return to its charger afterwards. Define

this task as a search problem and give all the components necessary to solve it using either uninformed

or informed search.

A picture containing shape

Description automatically generated

Start : (4)

State :

Action : Move Left, Right, Up, Down, must cover dirt before going to charger

Path Cost : 1 for any move to clean , 0 for dirt or charger

Goal : Charger (7)

We do a Uniform Cost search.

Diagram

Description automatically generated

Optimum path is 4->5->2->3->6->9->8->7, total cost : 4

7. Given your problem definition, which technique would you use? What would change if the robot had to

recharge after cleaning a dirty cell? Describe in words how this would change your problem definition.

I would use lowest cost search , ie Uniform cost search.

The goal will be changed to go 7 cell after 5 and 3.

Diagram

Description automatically generated