

## Lab 7: Motion Planning - Written Answers

Instructor: GROSU

Name: **TEAM 3** (Adelmann, Brantner, Lukitsch, Pintaric)

This lab and all related course material on [F1TENTH Autonomous Racing](#) has been developed by the Safe Autonomous Systems Lab at the University of Pennsylvania (Dr. Rahul Mangharam). It is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#). You may download, use, and modify the material, but must give attribution appropriately. Best practices can be found [here](#).

**Course Policy:** Read all the instructions below carefully before you start working on the assignment, and before you make a submission. All sources of material must be cited. The University Academic Code of Conduct will be strictly enforced.

**THIS IS A GROUP ASSIGNMENT.** Submit one from each team.

## 1 Part A: Written assignment

0	1	2	3	4	5	6	GOAL
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
START	41	42	43	44	45	46	47

Figure 1: Grid World

### 1.1 Grid world planning with Dijkstra's

Using figure 1, write out steps for Dijkstra's algorithm (**8-connected**, assume uniform cost for each action). At each step, list the grid cells in the open set with their running cost and the grid cells in the visited set. Write the final path found as a list of grid cell ids.

#### Solution:

On the following pages the calculation of the Dijkstra-Algorithm is shown. Our found path has a length of 7 and was found by always selecting the cell with the lowest index first, if there were 2 cells with the same cost.

The found path is:  $S \Rightarrow 33 \Rightarrow 26 \Rightarrow 27 \Rightarrow 20 \Rightarrow 21 \Rightarrow 14 \Rightarrow G$

The set of the **visited** cells can be seen in the **Set** column, for each step all the upwards entries of the column have been visited previously. Also the different sets with the currently **open** cells and their current costs can be seen in the vertical line of each step. If they have already been visited, they have no cost entry, if there has not been a path found the cost is listed with  $\infty$  and the currently selected cell is listed in bold font, if it is selected and visited in the current step.

In the last column **Pre** the previous cell of the selected cell is listed, so we can traverse the found tree backwards from a certain goal-cell. The found path from Start to Goal is underlined in the **Pre** column.

Step	S	41	42	43	44	45	46	47	32	33	38	39	24	25	26	27	28	29	31	20	21	22	12	14	15	0	1	2	3	4	5	6	G	Sel	Prev		
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	S	-			
1		1	∞	∞	∞	∞	∞	∞	1	1	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	32	S			
2		1	∞	∞	∞	∞	∞	∞		1	∞	∞	2	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	33	S			
3		1	2	∞	∞	∞	∞	∞			∞	∞	2	2	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	41	S			
4			2	∞	∞	∞	∞	∞			∞	∞	2	2	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	24	32			
5			2	∞	∞	∞	∞	∞			∞	∞		2	2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	25	32			
6			2	∞	∞	∞	∞	∞			∞	∞			2	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	26	33			
7			2	∞	∞	∞	∞	∞			∞	∞				3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	42	33			
8				3	∞	∞	∞	∞			∞	∞				3	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	27	26			
9				3	∞	∞	∞	∞			∞	∞					4	∞	∞	4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	43	42			
10					4	∞	∞	∞			∞	∞				4	4	∞	∞	4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	20	27			
11					4	∞	∞	∞			∞	∞				4	5	∞	∞		5	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	28	27			
12					4	∞	∞	∞			∞	∞					5	∞	∞		5	∞	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	44	43			
13						5	∞	∞			∞	∞					5	∞	∞		5	∞	5	∞	5	∞	∞	∞	∞	∞	∞	∞	12	20			
14						5	∞	∞			∞	∞					5	∞	∞		5	∞		∞	∞	∞	∞	∞	6	6	6	∞	21	20			
15						5	∞	∞			∞	∞					5	∞	∞			6		6	∞	∞	∞	∞	6	6	6	∞	29	20			
16						5	∞	∞			6	∞						∞	∞			6		6	∞	∞	∞	∞	6	6	6	∞	45	44			
17							6	∞			6	∞						∞	∞			6		6	∞	∞	∞	∞	6	6	6	∞	3	12			
18							6	∞			6	∞						∞	∞			6		6	∞	∞	∞	7		6	6	∞	4	12			
19							6	∞			6	∞						∞	∞			6		6	∞	∞	∞	7			6	∞	5	12			
20							6	∞			6	∞						∞	∞			6		6	∞	∞	∞	7				∞	14	21			
21							6	∞			6	∞						∞	∞			6			7	∞	∞	7				7	∞	22	21		
22							6	∞			6	∞						7	7					7	7	∞	∞	7				7	7	33	29		
23							6	∞				∞							7	7					7	7	∞	∞	7				7	7	46	45	
24								7				7							7	7					7	7	∞	∞	7				7	7	2	3	
25								7				7							7	7					7	∞	8					7	7	6	5		
26								7				7							7	7					7		8						7	7	15	14	
27								7				7							7	7						8							7	7	31	22	
28								7				7														8								7	7	39	46
29								7																		8								7	7	47	46
30																										8								7	G	14	
31																										8								1	2		
32																										9								0	1		

## 1.2 Grid world planning with A\*

Using figure 1, write out steps for A\* (**8-connected**, assume uniform cost for each action). At each step, list the grid cells in the open set with their f-value (use Manhattan distance to the goal as the heuristic function) and the grid cells in the visited set. Write the final path found as a list of grid cell ids.

### Solution:

On the following page the two possible runs of the A\* algorithm are shown, the algorithms were aborted after the Goal was visited, as its cost cannot be improved afterwards.

One of these is optimal (second one) and the first one is not optimal, since it returns not an optimal value for the path from Start to Goal. This is, because the heuristic is not **Admissible**, this means, it overestimates the distance from the current cell to the goal. Therefore, depending on the choice at step 5 (cell 12 or 21, both have a cost function **f** of 9), we either get an optimal path or not.

The first, not optimal path found is:

$S \Rightarrow 33 \Rightarrow 26 \Rightarrow 27 \Rightarrow 20 \Rightarrow 12 \Rightarrow 5 \Rightarrow 6 \Rightarrow G$  (cost 8)

The second and optimal path found is:

$S \Rightarrow 33 \Rightarrow 26 \Rightarrow 27 \Rightarrow 20 \Rightarrow 21 \Rightarrow 14 \Rightarrow G$  (cost 7)

Like with the Dijkstra algorithm, it is possible to see the set of visited nodes in the **Set** column upwards and the current costs of the **open** nodes in the horizontal lines of the table. In the first row of the table, the cell-ID and its heuristic value (Manhattan-Distance to the goal) is shown.

Step	S	41 (11)	42 (10)	43 (9)	44 (8)	45 (7)	46 (6)	47 (5)	32 (11)	33 (10)	38 (5)	39 (4)	24 (10)	25 (9)	26 (8)	27 (7)	28 (6)	29 (5)	31 (3)	20 (5)	21 (4)	22 (3)	12 (4)	14 (2)	15 (1)	0 (7)	1 (6)	2 (5)	3 (4)	4 (3)	5 (2)	6 (1)	G (0)	Sel	Pre	
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	S	-	
1		1+11	∞	∞	∞	∞	∞	∞	1+11	1+10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	33	S	
2		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9	2+8		∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	26	33	
3		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9		3+7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	27	26	
4		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞	4+5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	20	27	
5		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞		5+4	∞	5+4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	12	20	
6		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞		5+4	∞		∞	∞	∞	∞	∞	∞	6+4	6+3	6+2	∞	∞	5	12
7		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞		5+4	∞		7+2	∞	∞	∞	∞	∞	6+4	6+3		7+1	∞	6	5
		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞		5+4	∞		7+2	∞	∞	∞	∞	∞	6+4	6+3		8+0	G	6	
0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	S	-	
1		1+11	∞	∞	∞	∞	∞	∞	1+11	1+10	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	33	S	
2		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9	2+8		∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	26	33	
3		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9		3+7	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	27	26	
4		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞	4+5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	20	27	
5		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	∞	∞	4+5	5+4	∞	5+4	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	21	20	
6		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	6+5	∞	4+5		6+3	5+4	6+2	∞	∞	∞	∞	∞	∞	∞	∞	14	21		
7		1+11	2+10	∞	∞	∞	∞	∞	1+11		∞	∞	2+10	2+9			4+6	6+5	∞	4+5		6+3	5+4		7+1	∞	∞	∞	∞	∞	7+2	7+1	7+0	G	-	