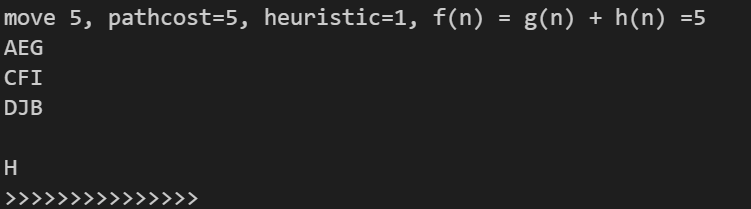
CSCE 420 - Spring 2022 Assignment A1

**Describe your heuristic**

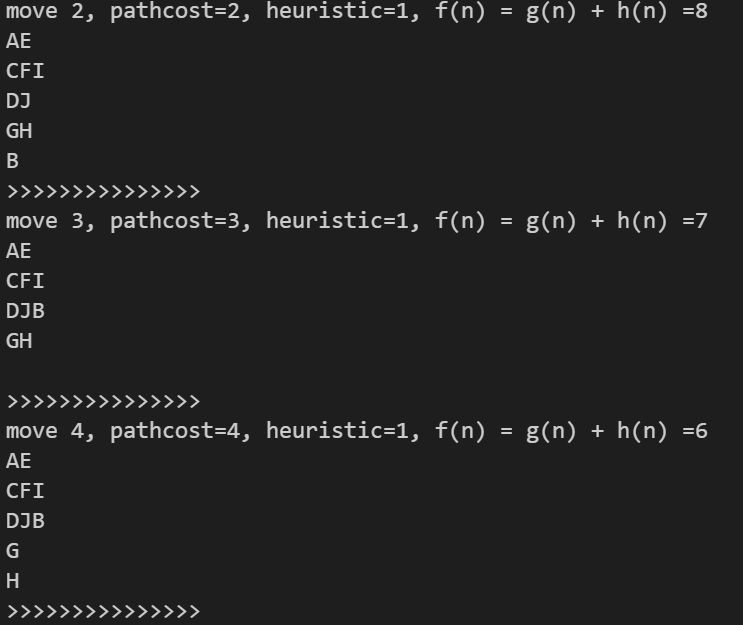
Simplify first step of heuristic is to get the distance between stack box from current state and goal state. We add a double multiplier to heuristic in the case the difference with current state having more nodes compare with the goal state. Which we then have the blocks compared individually until match is founded in both positions. In the case that blocks are not in the correct position within the stack when the top block is in correct position, we add 4 weights (or bad points) to the heuristic. When blocks just doesn’t match up between goal and current state we added a weight of 2, also do the same when the block is in correct stack, however not in the right place.

2 example to calculate H(n)

**Goal State**



We are going to look at move 2 and 4



Where we can see that

Move 2:

AE

CFI

DJ

are in the designated place which produced 0 weight

we add 2 (B) + 2(G incorrect stack)\*2 + 2 (H incorrect stack)= 8

Move 4:

AE

CFI

DJB

G

H

Are correct

We add

2(G) + 4 (incorrect column for AEG) = 6

Resources:

<https://www.cs.huji.ac.il/w~ai/projects/2012/SolvingBlocksWorldProblems/>

**Result A\* with my best heuristic Table:**

statistics: probs\probA03.bwp method Astar H1 planlen 3 iter 2 maxq 7

statistics: probs\probA04.bwp method Astar H1 planlen 4 iter 4 maxq 11

statistics: probs\probA05.bwp method Astar H1 planlen 5 iter 5 maxq 16

statistics: probs\probA06.bwp method Astar H1 planlen 6 iter 11 maxq 26

statistics: probs\probA07.bwp method Astar H1 planlen 7 iter 7 maxq 16

statistics: probs\probA08.bwp method Astar H1 planlen 8 iter 13 maxq 28

statistics: probs\probA09.bwp method Astar H1 planlen 9 iter 14 maxq 33

statistics: probs\probA10.bwp method Astar H1 planlen 10 iter 61 maxq 107

statistics: probs\probA11.bwp method Astar H1 planlen 11 iter 25 maxq 43

statistics: probs\probB03.bwp method Astar H1 planlen 3 iter 2 maxq 31

statistics: probs\probB04.bwp method Astar H1 planlen 4 iter 3 maxq 50

statistics: probs\probB05.bwp method Astar H1 planlen 5 iter 4 maxq 60

statistics: probs\probB06.bwp method Astar H1 planlen 6 iter 5 maxq 59

statistics: probs\probB07.bwp method Astar H1 planlen 7 iter 17 maxq 179

statistics: probs\probB08.bwp method Astar H1 planlen 9 iter 19 maxq 166

statistics: probs\probB09.bwp method Astar H1 planlen 9 iter 17 maxq 172

statistics: probs\probB10.bwp method Astar H1 planlen 10 iter 11 maxq 131

statistics: probs\probB11.bwp method Astar H1 planlen 9 iter 15 maxq 169

statistics: probs\probB12.bwp method Astar H1 planlen 9 iter 8 maxq 120

statistics: probs\probB13.bwp method Astar H1 planlen 14 iter 600 maxq 4325

statistics: probs\probB14.bwp method Astar H1 planlen 14 iter 70 maxq 730

statistics: probs\probB15.bwp method Astar H1 planlen 16 iter 140 maxq 1157

statistics: probs\probB16.bwp method Astar H1 planlen 18 iter 536 maxq 4260

statistics: probs\probB17.bwp method Astar H1 planlen 16 iter 36 maxq 422

statistics: probs\probB18.bwp method Astar H1 planlen 12 iter 18 maxq 199

statistics: probs\probB19.bwp method Astar H1 planlen 14 iter 99 maxq 778

statistics: probs\probB20.bwp method Astar H1 planlen 16 iter 320 maxq 2690

**Are solution path lengths, number of iterations, max queue size, and other metrics reasonable? How do they scale up with harder problems?**

The obtained data seems to reasonably grow and scale upwards as the problem gets more difficult. There are certain problems that had a spike in more iteration and max queue along the way, but likely causes are from the A\* algorithm implemented is not suit for those cases.