

Critical Thinking Module 4

Michael Stewart

Colorado State University Global

CSC510-1: Foundations of Artificial Intelligence

Professor Bingdong Li

April 28, 2024

Tower of Hanoi

In this module, I created a Python script that uses SimpleAI's library to search through all the possibilities of a size 3 Hanoi Tower. This is an interesting program because it uses an A* search with a heuristic function to find the optimal solution path. One of the struggles that I had with this was creating a heuristic function. I had some good ideas about adding points based on which discs were in what position on the third pole, but I had some issues with the priority queue. To solve this problem, I used a heuristic function where the program gets a better score if it has more discs on the tower, not necessarily taking into account which discs were in each position. This solution immediately worked and gave the optimal path for a size 3 Hanoi tower. If I were to spend more time on this program, then I would try and adapt my program to do even taller Hanoi towers. I based my implementation off of the `eight_puzzle` sample program SimpleAI uses as a tutorial. This game also used a list of lists and gave the SimpleAI data through a string. I was able to learn about how they handle lists of lists and use them for my Hanoi tower code. I used an A* search because I wanted to implement a heuristic function so that the program could intelligently know how close it is to a full tower. This is space efficient because it is able to explore optimal paths first rather than multiple infinite paths and have to store their information. These advantages are great, but there are some disadvantages like how dependant it is on my heuristic function for its efficiency.

```

1 from __future__ import print_function
2 from sipmlal.search import searchproblem, astar
3 import copy
4
5
6
7 def list_to_string(list_):
8     return '\n'.join(['\n'.join(map(str, row)) if isinstance(row, list) and len(row) >= 3 else '\n'.join(map(str, [0] * (3 - len(row)) + row)) if isinstance(row, list) else str(row) for row in list_])
9
10
11 def string_to_list(string):
12     return [[int(x) for x in row.split('.') if x != '0'] if row.strip() and row != '0-0-0' else [] for row in string.split('\n')]
13
14 GOAL = [[1,2,3]]
15 INITIAL = [[1,2,3]]
16 GOAL = list_to_string(GOAL)
17 INITIAL = list_to_string(INITIAL)
18
19
20 class HelloProblem(searchproblem):
21     def actions(self, rows):
22         actions = []
23         state = string_to_list(rows)
24         if state[0]:
25             if not (state[1]):
26                 temp = copy.deepcopy(state)
27                 temp[1].insert(0, temp[0][0])
28
29
30 PS C:\Users\Michael\Documents\GitHub\PythonProjects\CSC510\Week4\Critical Thinking 4> python3 .\Tower_of_Hanoi.py
31 0-0-0
32 0-0-0
33 1-2-3
34 [(None, '1-2-3\n0-0-0\n0-0-0'), ('0-2-3\n0-0-0\n0-0-1', '0-2-3\n0-0-0\n0-0-1'),
35 ('0-0-3\n0-0-2\n0-0-1', '0-0-3\n0-0-2\n0-0-1'), ('0-0-3\n0-1-2\n0-0-0', '0-0-3\n0-1-2\n0-0-0'),
36 ('0-0-0\n0-1-2\n0-0-3', '0-0-0\n0-1-2\n0-0-3'), ('0-0-1\n0-0-2\n0-0-3', '0-0-1\n0-0-2\n0-0-3'),
37 ('0-0-1\n0-0-0\n0-2-3', '0-0-1\n0-0-0\n0-2-3'), ('0-0-0\n0-0-0\n1-2-3', '0-0-0\n0-0-0\n1-2-3')]

```

Output:

PS C:\Users\Michael\Documents\GitHub\PythonProjects\CSC510\Week4\Critical Thinking

4> python3 .\Tower_of_Hanoi.py

0-0-0

0-0-0

1-2-3

[(None, '1-2-3\n0-0-0\n0-0-0'), ('0-2-3\n0-0-0\n0-0-1', '0-2-3\n0-0-0\n0-0-1'),

('0-0-3\n0-0-2\n0-0-1', '0-0-3\n0-0-2\n0-0-1'), ('0-0-3\n0-1-2\n0-0-0', '0-0-3\n0-1-2\n0-0-0'),

('0-0-0\n0-1-2\n0-0-3', '0-0-0\n0-1-2\n0-0-3'), ('0-0-1\n0-0-2\n0-0-3', '0-0-1\n0-0-2\n0-0-3'),

('0-0-1\n0-0-0\n0-2-3', '0-0-1\n0-0-0\n0-2-3'), ('0-0-0\n0-0-0\n1-2-3', '0-0-0\n0-0-0\n1-2-3')]

References

Search algorithms — simpleai 0.8.2 documentation. (n.d.). Retrieved April 28, 2024, from

https://simpleai.readthedocs.io/en/latest/search_problems.html

Sharda, R., Delen, D., & Turban, E. (2024). *Business Intelligence, analytics, data science, and ai*. Pearson.

simpleai-team. (n.d.). *simpleai/samples/search/eight_puzzle.py at master* ·

simpleai-team/simpleai. GitHub. Retrieved April 28, 2024, from

https://github.com/simpleai-team/simpleai/blob/master/samples/search/eight_puzzle.py