Assignment 2

Andrew B. Moore

Due 9/23/2024

#usf/cs/ai/assignments

1. Algorithms

Algorithm	Time Complexity	Space Complexity	Complete?	Optimal?
BFS	O(b^d)	O(b^d)	Yes	Yes
UCS	O(b^d)	O(b^d)	Yes	Yes
DFS	O(b^m)	O(bm)	No	No
DLS	O(b^l)	O(bl)	No	No
IDS	O(b^d)	O(bd)	Yes	No
A*	O(b^d)	O(b^d)	Yes	Yes (with proper heuristic)

- `b`: Branching factor (the average number of successors per state).
- `d`: Depth of the shallowest goal node.
- `m`: Maximum depth of the search tree.
- `l`: Depth limit in DLS.

2. Mars_Planner

See `mars_planner_states.pdf`

3. Path-finding (A*)

`sld`:

```
A* generated 34 states
[(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (2,7), (3,7), (3,6), (3,5),
(3,4), (4,4), (5,4), (5,5), (6,5), (6,6), (7,6), (8,6), (8,7), (8,8)]
```

`h1`:

```
A* generated 32 states
[(1,1), (1,2), (1,3), (1,4), (1,5), (1,6), (1,7), (2,7), (3,7), (3,6), (3,5),
(3,4), (4,4), (5,4), (5,5), (6,5), (6,6), (7,6), (8,6), (8,7), (8,8)]
```

4. Constraints

Output:

```
Antenna 1: f2
Antenna 2: f1
Antenna 3: f3
Antenna 4: f2
Antenna 5: f3
Antenna 6: f2
Antenna 7: f1
Antenna 8: f3
Antenna 9: f1
```

5. Deep Blue vs. AlphaZero

a.) Deep Blue was the result of several innovations, including a massively parallel architecture, advanced search (including quiesence search, iterative deepening, and transposition tables), hardware optimizations, and an “opening book” (to my understanding, this is essentially a massive corpus of historical data). The most transferrable of these to other tasks are Deep Blue’s parallel architecture, search, and hardware optimizations, as these are all relatively general—features like the “opening book” are more specific to chess/game-playing, and likely aren’t directly generally applicable, though I’m almost certain *someone*, *somewhere* has found a way to apply them to other problems.

b.) AlphaZero's advantage over Stockfish and other models is due to significant differences in architecture, most notably in search strategies and evaluations. In terms of search, AlphaZero uses a Monte Carlo Tree Search, while Stockfish uses alpha-beta pruning—the latter evaluates far more positions than AlphaZero's method, requiring longer compute time. AlphaZero also has a "selective focus" mechanism, where it narrows down its search as it progresses, in contrast, Stockfish evaluates possible moves and states in a much wider fashion, which both takes longer and is less accurate.