Assignment 2: Search

Andrew Diep

Question 1

Algorithm	Time Complexity	Space Complexity	Complete?	Optimal?
BFS	O(b^(d+1)), where b is the branching factor, and d is the depth of the solution.	O(b^(d+1)), where b is the branching factor, and d is the depth of the solution.	Yes	Yes
UCS	O((E+V)logV), where E is the number of edges and V is the number of vertexes	O((E+V)), where E is the number of edges and V is the number of vertexes	Yes	Yes
DFS	O(b^n), where n is the depth of the search tree	O(bn)	Yes on a finite graph No if search space is infinitely large, or we can't check for repeated states	No
DLS	O(b^L), where b is the branching factor and L is the depth limit	O(bL), where b is the branching factor and L is the depth limit	Yes if the depth limit is set to a value greater than or equal to the depth of the shallowest goal node in the search space	Yes

			No if it’s less than the depth limit	
IDS	O(b^d), where b is the branching factor and d is the depth of the shallowest goal node	O(bd), where b is the branching factor and d is the depth of the shallowest goal node	Yes	Yes
A*	O(b^d), where b is the branching factor and d is the depth of the shallowest goal node	O(b^d), where b is the branching factor and d is the depth of the shallowest goal node	Yes	Yes

Question 2

```
a.
None

b.
None

c.
No goal found
Number of states generated: 3
None

d.
No goal found
Number of states generated: 3
None

e.
```

```
BFS
Goal found
Number of states generated: 26
(Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True
Holding Tool? False, 'charge')

DFS
Goal found
Number of states generated: 9
(Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True
Holding Tool? False, 'charge')

DLS
Goal found at depth 7
Number of states generated: 9
Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True
Holding Tool? False

f.
BFS
Goal found
Number of states generated: 5
Goal found
Number of states generated: 6
Goal found
Number of states generated: 23
Location: battery
```

```
Sample Extracted?: True
Holding Sample?: False
Charged? True
Holding Tool? False

DFS
Goal found
Number of states generated: 4
Goal found
Number of states generated: 11
Goal found
Number of states generated: 7
Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True
Holding Tool? False

DLS
Goal found at depth 2
Number of states generated: 4
Goal found at depth 9
Number of states generated: 11
Goal found at depth 4
Number of states generated: 7
Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True
Holding Tool? False
```

f) continued

The number of states generated for each subproblem is typically smaller compared to solving the entire problem at once. However, when combined, the total number of states generated across all subproblems exceeds the number generated if the problem were solved in one go.

Question 3

sld heuristic
Number of states generated: 32

h1 heuristic
Number of states generated: 32

Question 4

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

Question 5

a) What were the engineering advances that led to Deep Blue's success? Which of them can be transferred to other problems, and which are specific to chess?

The engineering advances that led to Deep Blue’s success include specialized hardware, parallel processing, alpha-beta pruning optimization, evaluation function, as well as the opening books and endgame databases.

Specialized hardware can be trasnffered to other problems, as with the custom chess chips, the idea of using specialized hardware for specific tasks can be applied to various fields requiring high-speed computation, such as image processing, cryptography, or real-time data analysis.

Parallel processing can be transferred to other problems, as it is widely applicable in many areas, including scientific simulations, large-scale data processing, and machine learning. Advances in multi-core processors and distributed computing build on this concept.

Alpha-beta pruning optimization can be transferred to other problems, as it is used in various artificial intelligence applications beyond chess, such as game theory, decision-making algorithms, and resource management.

Evaluation functions and scoring mechanisms are crucial in many AI applications, such as recommendation systems, financial forecasting, and natural language processing. But in Deep Blue, it was tailored specifically to chess.

The opening books was specifically designed based on chess opening theory, which is unique to chess and does not directly transfer to other problems. The same goes for using endgame databases.

b) AlphaZero is compared to a number of modern game-playing programs, such as StockFish, which work similarly to Deep Blue. The paper shows that AlphaZero is able to defeat StockFish even when it is given only 1/100 of the computing time. Why is that? Please frame your answer in terms of search and the number of nodes evaluated.

StockFish uses alpha-beta pruning. Despite its efficiency improvements through various pruning strategies, it still relies on exploring a vast number of nodes to achieve its evaluation.

AlphaZero utilizes Monte-Carlo Tree Search (MCTS) combined with a deep neural network. It focuses computational resources on the most promising moves, guided by the neural network's evaluations, which can prioritize high-potential moves and avoid exhaustive search.

Stockfish employs a handcrafted evaluation function that uses detailed feature vectors and domain-specific knowledge. It typically evaluates a large number of nodes but can be limited by search depth due to the computational cost of evaluating each node thoroughly. Even with advanced pruning techniques, it often explores a broader but shallower search tree.

AlphaZero uses a neural network to evaluate positions. It focuses on fewer nodes by selectively expanding the most promising parts of the game tree. The neural network helps AlphaZero evaluate potential positions more effectively, allowing it to achieve high-quality assessments with fewer nodes explored.

This means that AlphaZero's computational resources are concentrated on the most crucial parts of the game tree, enabling deeper and more relevant exploration within the same time constraints, allowing it to beat StockFish.