David Shubov

Q1:

Algorithm	Time Complexity	Space Complexity	Complete?	Optimal?
BFS	O(b^d)	O(b^d)	Yes	Yes
UCS	O(b^d), but time influenced by cost instead of depth	O(b^d)	Yes	Yes
DFS	O(b^n)	O(bn)	Yes, on a finite graph where we can check for repeated states	No
DLS	O(b^{limit})	O($b\ *\ limit$)	Yes, if the limit is sufficient	No
IDS	O(b^d)	O(bn)	Yes	Yes
A*	O(b^d), depends on heuristic	O(b^d)	Yes, under certain conditions, like heuristic is admissible	Yes, under certain conditions, like heuristic is admissible

Q2 and Q3:

States	BFS	DFS	DLS
Battery goal	3	5	—
Mission complete/tools for extraction	39	16	depth=5, 29 depth=15, 16
Decomposition	23	9	—
A* (Q3)	H1=32	SLD=32	—

Q5:

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?

Several engineering advances led to Deep Blue’s success. One that gets mentioned a lot in the first article is parallel processing. The software was redesigned to handle parallel search better. The system had over 500 processors ready to do the heavy lifting. But, the researchers pointed out that these advancements also came with challenges. For example, no one had ever worked with systems that “searched orders of magnitude fewer than Deep Blue” (4).

Another big advancement was the algorithms, like dual-credit and the extended book in Deep Blue. These gave it a huge set of knowledge and access to the Grandmaster game database when there was no opening book information.

Some of the advancements were specific to chess, like the new chess chip and move generation algorithms. However, others weren’t limited to chess. For instance, alpha-beta pruning can be used in many two-player games, and depth-limited searching, which we discussed in class and is part of this assignment, applies to various decision-making problems.

A really important advancement to highlight is mass parallelism. Nowadays, systems are getting larger and more complex and need more processing power to run effectively. This is especially true in fields with heavy mathematical computation and machine learning.

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.

From what I understand regarding search, AlphaZero evaluates positions based on a deep neural network, combined with a Monte-Carlo search. These two algorithms complement each other since the neural network could introduce “spurious approximation errors” (12) but the MCTS helps to cancel out the errors in a large tree. Additionally the MCTS focused on high move probability and value, helping the neural network prioritize more promising paths.

These mechanisms allowed AlphaZero to conduct reinforcement learning by performing searches that were a series of simulated “self-play” games. Because AlphaZero relied on self-play rather than something like heuristics from Stockfish, it could perform more intelligent and strategic searches, reducing redundant or unfavorable node exploration, leading to quicker wins.