Question 1:

Algo	TC	SC	Complete?	Optimal?
BFS	O(V+E)	O(V)	Yes when branching if limited	Yes If there is equal cost between nodes
UCS	O(V+E)	O(V)	Yes, if search graph is limited	Yes, selects lowest cost path
DFS	O(V+E)	O(V)	No not if nodes are unlimited	No, shortest path non guaranteed
DLS	O(V+E)	O(l) limit	No, it can miss the goal located past limit	No, shortest path non guaranteed
IDS	O(V+E)	O(BD) d=depth of shortest solution	Yes, if graph is limited	Yes, under conditions like uniform step costs
A*	O(V+E)	O(V)	Yes if heuristic is admissible and motonic as well as on graphs that are limited	Yes if heuristic is admissible and motonic as well as on graphs that are limited

Question 2:

a. Some of the engineering advances that helped solidify Deep Blue's success were things like complex evaluation function, and effective use of a Grandmaster game database. Because of these functions, this allowed Deep Blue to be able to use more complex functions specifically for assessing chess positions. Detailed factors like king safety, control of the center, etc. The specific heuristics it used were tailor-made for chess making it a worthy opponent. Additionally, It was able to take advantage of the database of grandmaster games so things like opening moves and endgame positions from grandmasters were just another addition to its arsenal. Now while these are specific to chess, the other advances that were applicable to other problems were functions like search algorithms, selective search, parallel processing hardware and more. It was able to evaluate millions of positions per second, use selective search to focus on the most effective moves along with the help of algorithms(alpha-beta).

b. One key difference between stockfish and alphazero is that alphazero utilizes something called a monte-carlo search tree algorithm. The way it works is that each search consists of a series of simulated games against itself that traverse a tree from root to leaf.

Basically, it uses a deep neural network to evaluate positions and predict the most likely outcome of games from the positions. It evaluates fewer nodes per second than traditional engines like the one in stockfish which uses more of a brute-force search approach which scans a vast library of nodes in a search tree to determine its best move. This is computationally expensive as it evaluates hundreds of millions of nodes per second. Alphazero focuses on a deep search of a narrow set of positions. Alphazero is focused more on quality rather than quality like stockfish. It is like breadth-first search versus depth-first search. Integration of deep learning with ai search techniques proves significant here.