Question 1:

Algorithm	**Time Complexity**	**Space Complexity**	**Complete?**	**Optimal?**
BFS	$O(b^{d+1})$	$O(b^{d+1})$	Yes	Yes
UCS	$O(b^{(1+C/m)})$	$O(b^{(1+C/m)})$	Yes	Yes
DFS	$O(bd)$	$O(b^{d})$	Yes, Only on finite search spaces.	No
DLS	$O(b^{L})$	$O(bL)$	No	No
IDS	$O(b^{d})$	$O(bd)$	Yes, Only on finite search spaces.	Yes, when edge costs are the same
A*	$O(E)$	$O(N)$	Yes	Yes, Only in cases where the heuristic is admissible.

B = branches; D = depth; L = limit; C = cost; M = max cost; E = edges; N = nodes

Question 2 Counts:

Part 3)
BFS: count = 4
DFS: count = 2

Part 5)
BFS : count = 46
DFS: count = 63
DLS: ran with limit = 17, count = 60

Part 6)
moveToSample: count = 3
removeSample: count = 28
returnToCharger: count = 7

Question 3 Counts:

A*: count = 31
UCS: count = 31

Question 5 Written Answers:

Question A)

There are a few Engineering advances that led to Deep Blue's success. One of the advances that caught my attention was the fact they had two different searches, the hardware search and the software search. And both searches operated differently, with software searching the roots of the tree and hardware searching the leaves of the tree. Having these work together in one system without causing conflicts to each other must have been a big advancement. As well, they also implanted a "No progress" mechanism, which decided to play good moves earlier in the game instead of holding them for later. Another advancement that was made was their implementation of parallel search. They were even able to address their load balancing and master overload issues through this search implementation. This as well allowed them to have a large search capability. The next advancement was their pawn tables. They had several different features that went into classifying the pawns on the board into how they were laid out, and what their openings and threats looked like, and the king's safety levels.

Of the advancements discussed above, not all of them can be directly transferred to other problems not related to chess. I think the "no progress" mechanism and the pawn tables are fairly chess specific, as they relate to the logistics and functioning of the chess games. However, I think that the differing searches in hardware and software could be applied elsewhere, as well as the parallel searching which could help resolve issues in some other problems as well.

Question B)

AlphaZero is able to defeat StockFish even with a significantly smaller computing time because of it's different take to the search algorithm. StockFish uses the beta-alpha search of visiting all possibilities of moves, and going to all nodes of the tree. However, AlphaZero uses a different approach through the use of a neural network and a Monte Carlo approach of focusing on promising variations of possibilities. This eliminates the time spent because they are not visiting every node, only ones that have a level of promise.