1.

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	Yes
A*	Depend on heuristic	O(b^d)	Yes	Yes

2. **Deep Blue vs AlphaZero**
    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?
        a. Deep blues success was driven by engineering advances such as custom built hardware allowed for close to 180 million chess moves per second, additionally the custom evaluation function that was human tuned. Some approaches that can be transferred to other problems is like training it with the opening books, that kind of concept of training the machines with existent human knowledge can be transferred to other problems, the ones that are specific to chess are the evaluation function those are less transferable to other problems.
    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.
        a. AlphaZero outperforms Stockfish, despite using 1/100 of computing time because Stockfish essentially uses search algorithms to find the best moves, this sometimes leads to ineffective searches and uses a lot of resources. In contrast, AlphaZero uses new technologies like deep reinforcement learning to guide its search focusing only on the most promising

3. **States generated without using problem decomposition**

Using sld
Goal found: (1,1)
counter: 24
(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)

Using h1
Goal found: (1,1)
counter: 32

(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)

States using problem decomposition
Goal found
(Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True, 'charge')
Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? False
Location: station
Sample Extracted?: True
Holding Sample?: False
Charged? False
Location: station
Sample Extracted?: True
Holding Sample?: False
Charged? False
Location: sample
Sample Extracted?: True
Holding Sample?: False
Charged? False
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? False
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? False
Location: station
Sample Extracted?: False
Holding Sample?: False
Charged? False
None
Goal found
(Location: battery

Sample Extracted?: True
Holding Sample?: False
Charged? True, 'move_to_battery')
Location: station
Sample Extracted?: True
Holding Sample?: False
Charged? True
Location: station
Sample Extracted?: True
Holding Sample?: False
Charged? True
Location: station
Sample Extracted?: True
Holding Sample?: False
Charged? True
Location: sample
Sample Extracted?: True
Holding Sample?: False
Charged? True
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? True
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? True
Location: battery
Sample Extracted?: False
Holding Sample?: False
Charged? True
Location: battery
Sample Extracted?: False
Holding Sample?: False
Charged? False
Location: station
Sample Extracted?: False
Holding Sample?: False
Charged? False
None
Sub-problem 1: Move to Sample
Goal found
(Location: sample
Sample Extracted?: False

Holding Sample?: False
Charged? False, 'move_to_sample')
Location: station
Sample Extracted?: False
Holding Sample?: False
Charged? False
None
Sub-problem 2: Remove Sample
Goal found
(Location: sample
Sample Extracted?: True
Holding Sample?: False
Charged? False, 'use_tool')
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? False
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? False
Location: station
Sample Extracted?: False
Holding Sample?: False
Charged? False
None
Sub-problem 3: Return to Charger
Goal found
(Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? True, 'charge')
Location: battery
Sample Extracted?: True
Holding Sample?: False
Charged? False
Location: sample
Sample Extracted?: True
Holding Sample?: False
Charged? False
Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? False

Location: sample
Sample Extracted?: False
Holding Sample?: False
Charged? False
Location: station
Sample Extracted?: False
Holding Sample?: False
Charged? False