

Eva DeThomas

Assignment 2 Results:

Question 1:

Algorithm	Time Complexity	Space Complexity	Complete?	Optimal?
BFS	$O(V + E)$	$O(V)$	Yes, guaranteed to find solution	Yes, assuming all actions have a uniform cost
UCS	$O((V+E)\log V)$	$O(V)$	Yes, guaranteed to find a solution	Yes, guaranteed to find correct solution
DFS	$O(b^n)$, where n is the depth of the search tree.	$(O(bn))$ Linear	Complete on a finite graph, if space is infinite or we can't search for repeated states then no.	No guarantee the first solution is the best
DLS	$O(b^L)$, where L is the pre-set limit of the search tree.	$(O(L))$	No, the limit allows for efficiency but not all nodes are explored	No, the first solution found will be return regardless of if it's the best or not
IDS	$O(b^d)$	$O(bd)$	Yes, will not stop until solution is found	Yes, guaranteed to find correct solution
A*	At worst: $O(b^d)$	$O(b^d)$	Depends on conditions, can be complete and optimal	Depends on conditions, can be complete and optimal

Question 2:

(All answers are post adding tool functionality)

Mission complete:

Final mission complete DFS result count: 19

Final mission complete BFS result count: 23
Final mission complete LDFS result count: 9 (limit is 6)

Partitioned tasks:

<function go_to_sample at 0x105854ca0> Final DFS result count: 12
<function go_to_sample at 0x105854ca0> Final BFS result count: 17
<function go_to_sample at 0x105854ca0> Final LDFS result count: 12 limit is 8

<function remove_sample at 0x105854d30> Final DFS result count: 7
<function remove_sample at 0x105854d30> Final BFS result count: 4
<function remove_sample at 0x105854d30> Final LDFS result count: 7 limit is 8

<function return_to_charger at 0x105854dc0> Final DFS result count: 11
<function return_to_charger at 0x105854dc0> Final BFS result count: 11
<function return_to_charger at 0x105854dc0> Final LDFS result count: 11 limit is 8

Even though we would expect to see less states visited in partitioned tasks, there are sometimes more because it had to do some extra searching to get to the proper state (backtracking). (Can run the program for more specific state data)