

Algorithm	Time Complexity	Space Complexity	Complete?	Optimal?
BFS	$O(b^d)$ b is the branching factor d is the depth of the solution	$O(b^d)$ Entire tree goes in memory	Yes	Yes
UCS	$O(b^{(c/e)})$ c is the optimal solution e is smallest step cost	$O(b^{(c/e)})$ Same as BFS	Yes	Yes
DFS	$O(b^n)$ b is branching factor n is depth of search tree	$O(bn)$ Only needs to store the current search branch	Yes unless the graph is infinitely large or we cannot check for repeated states	No, does not always find the best solution
DLS	$O(b^l)$ b is branching factor l is depth limit of search tree	$O(bl)$ Only needs to store the current search branch	Yes unless the solution is beyond the depth limit	No, does not always find the best solution
IDS	$O(b^d)$ Same as BFS	$O(bd)$ Only needs to store the current search branch	Yes	Yes
A*	$O(b^d)$ ^ In worst case Mostly depends on how good of a heuristic is used	$O(b^d)$ Stores all expanded nodes in memory.	Yes	Yes in cases where our heuristic is admissible (underestimates the true cost to the goal)