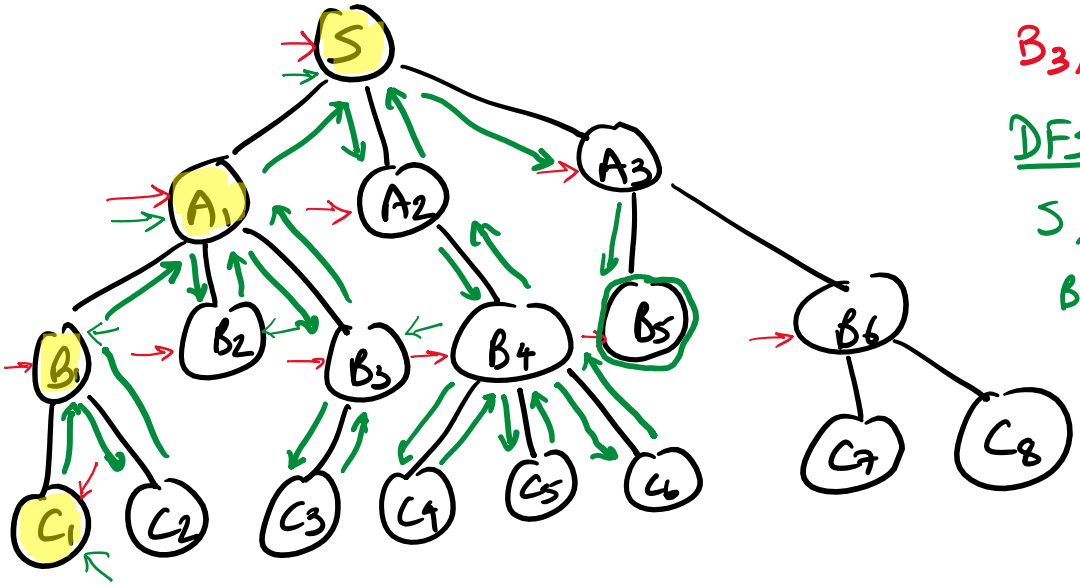
AI Uninformed Search Strategies

Saturday, September 14, 2024 1:00 PM

☆ DFS - Depth First search



BFS:

S, A_1, A_2, A_3, B_1, B_2, B_3, B_4, B_5

DFS:

S, A_1, B_1, C_1, C_2, B_2, B_3, C_3, A_2, B_4, C_4, C_5, C_6, A_3, B_5

Evaluation of DFS:

① Completeness :- No (If any of the branches has infinite depth & if goal state does not exist in that branch or the branches before it then DFS will never reach to goal state)

② Optimality :- Not Optimal. (consider two goal states C_1 & B_5 in previous graph)

③ Space Complexity :- DFS has to keep only the branch it is expanding in memory.

$O(n)$; n = depth of the current branch

④ Time Complexity :- $O(V+E)$ where V = no. of nodes

E = Total no. of edges in the graph

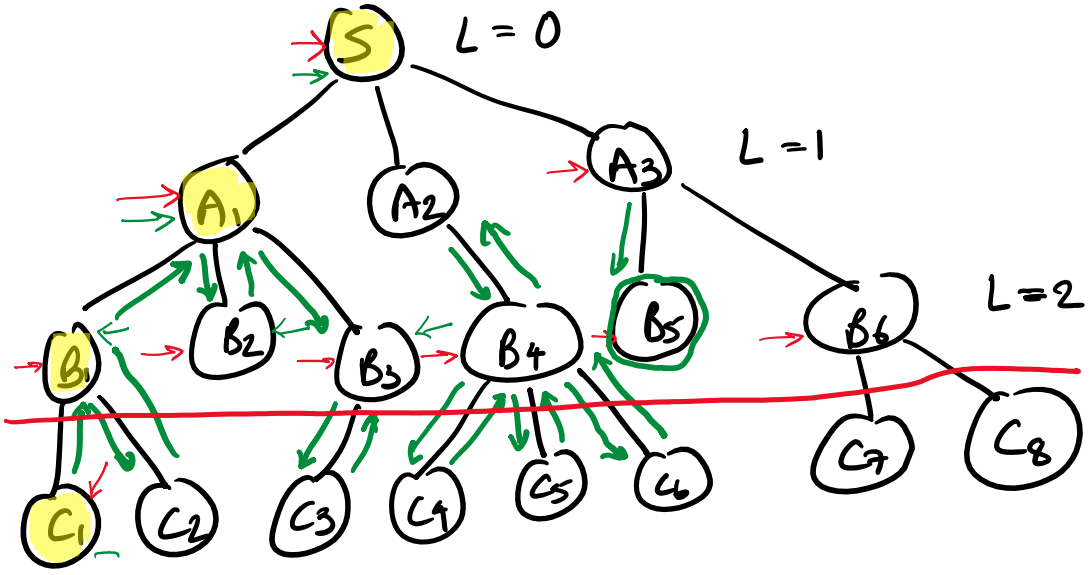
Time complexity: O(V+E) where V = no. of nodes (vertices), E = Total no. of edges in the graph

To be expanded : [S] [A3 A2 A1] [A3 A2 B3 B2 B1]

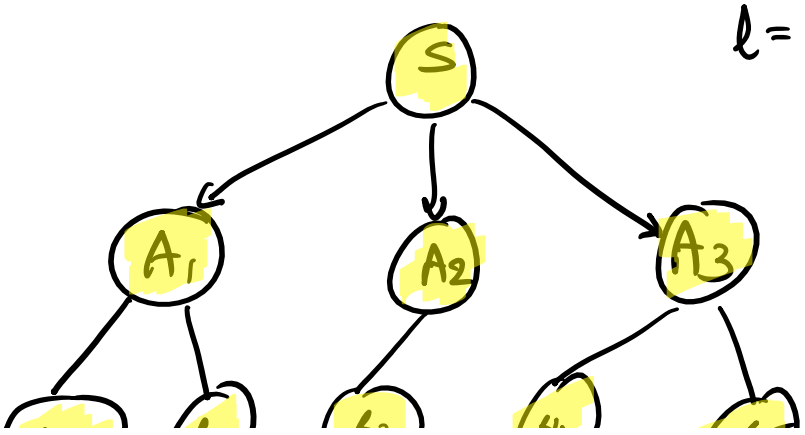
(LIFO)

[A3 A2 B3 B2 C2 C1] [A3 A2 B3 B2 C2] [A3 A2 B3 B2]

✡ Depth Limited Search :- L = 2



✡ Iterative Deepening DFS (ID-DFS) - First we set l = 0 in this algo & then after we keep incrementing l by 1 until we find the goal state.



goal state.

Advantage: Unlike to DFS this will never trap in an infinite loop plus, it will also now Complete as well as Optimal.

Disadvantage: Every time it starts over from the start state hence checking all the states for goal state unnecesarily over and over again.