21/1/2022

Depth Limit Search

Let fringe be a list containing initial state

Loop

if the fringe is empty return failure;

else

Node ← remove_first(Fringe)

if Node is goal

return path from initial state to Node

else if depth of Node == limit

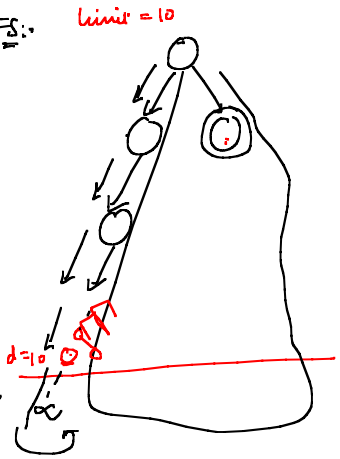
return cutoff;

else

Expand Node and add all generated Nodes to the front of fringe.

End Loop.

DFS:-



Complete :- Not ✓

optimality :- Not. ✓

Depth-First Iterative Deepening Search (DFID)

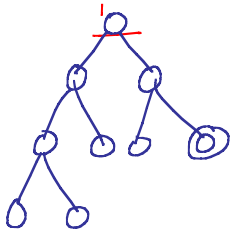
Approach :- first run DFS on depth 0 (Treat that Root node has no successor)

If no solution exists then do run DFS on depth 1

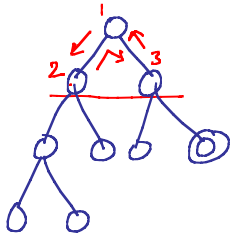
and so on.

```
C = 0 ;
Until solution is found do
        Run DFS with depth cutoff C ;
        C = C+1 ;
End
```

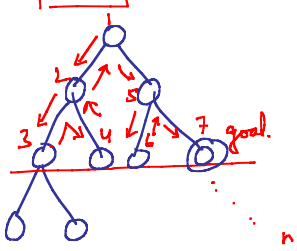
Depth C=0



C=1



C=2



Completeness :- Yes ✓

Optimal ✓

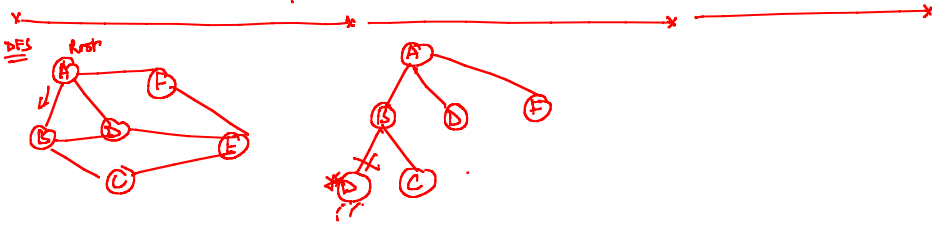
Time :- Worse than BFS/DFS

Space :- O(b.d) [same as DFS]

Time Complexity

BFS = $O(b^d)$

DFS = $O(b^m)$



Class Assignment 1