Assignment 1

20L-1122

Hassan Jawwad

6-B

Question 2

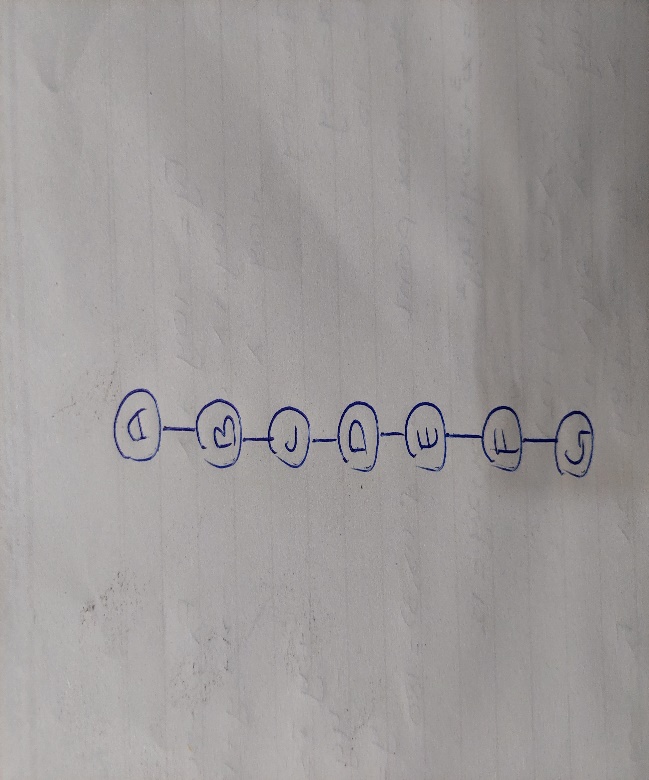
A)

If we have a state with a single branching factor and there is only one goal at depth n.

The DFS will find the goal node in O(n) steps whereas IDS will find the goal node in

1+2+3+…+n= O(n^2) steps.

For example:



Node A is the initial state and node G is the final(goal) state

DFS will explore the nodes

Level 1 to level 7: A to G

Total nodes explored: 7

ABCDEFG

Whereas IDS will explore the nodes

Level 1: A

Level 2: A,B

Level 3: A,B,C

Level 4: A,B,C,D

Level 5: A,B,C,D,E

Level 6: A,B,C,D,E,F

Level 7: A,B,C,D,E,F,G

Total nodes explored: Level 1: 28

A ,A,B,A,B,C,A,B,C,D,A,B,C,D,E,A,B,C,D,E,F,A,B,C,D,E,F,G

IDS have 4 times worse space complexity here

B)

A) Precise formulation of the task as a search problem:

Initial state:

One randomly selected piece.

Successor function:

For any open peg, add any piece type from the remaining types. We can add to open holes as well, but that isn’t necessary as all complete tracks can be made by adding to pegs.

For a curved piece, add in either orientation; for a fork, add in either orientation and (if there are two holes) connecting at either hole. Disallow any overlapping configurations, as this terminates hopeless configurations early.

Goal State:

All pieces used in a single connected track, no open pegs or holes, no over- lapping tracks.

Step cost:

One per piece

B) Suitable uniform search algorithm for this task:

DFS will be used in this case because of the big state space it will work best.

Whereas other search algorithms such as BFS or IDS may take many steps.

C) Removing the pieces makes the problem unsolvable:

If you remove the fork pieces, then you could not solve this problem. This would be due to the fact that when you use a fork to create a split of two tracks. The only way to then recombine these two tracks you would need to have a fork. In order to have an ending to the track you would therefore need forks.