Search

Iterative deepening DFS :→

DFS	BFS

Which Algorithm is better in terms of time and Space.

BFS : better in time Since $d < m$

↓

may be

↓

$b^d < b^m$

time / Space

↳ Which is most important.

Breadth First Search :→

1. If the frontier is too large
   * Keep the content in the frontier that are important to reach the goal.
   * Remove something (Randomly)

Beam Search

1. Constant Size Frontier (based on Space)
2. If the frontier becomes ~~too~~ large then prune/remove the worst nodes.

Optimal : X

Complete : X

NLP : Use of beam Search. (chatbot)

↓

Natural Language Processing

An Algorithm :—

1. good ~~is~~ in Space (Not Exponential)
2. good in time (Not Exponential on (m) but on (d))
3. Optimum
4. Complete.

Algorithm: Iterative deepening Search

function ITERATIVE-DEEPENING-SEARCH (problem)
returns a Solution, or failure.

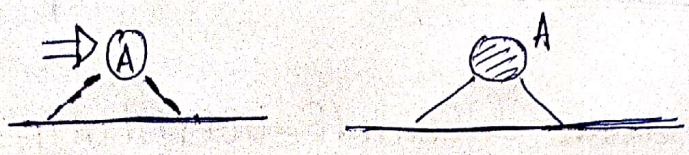
inputs: problem, a problem
for depth ← 0 to ∞ do
result ← DEPTH-LIMITED-SEARCH
(problem, depth)
if result ≠ cutoff then Return result.

DFS
↓
Stop at the middle.

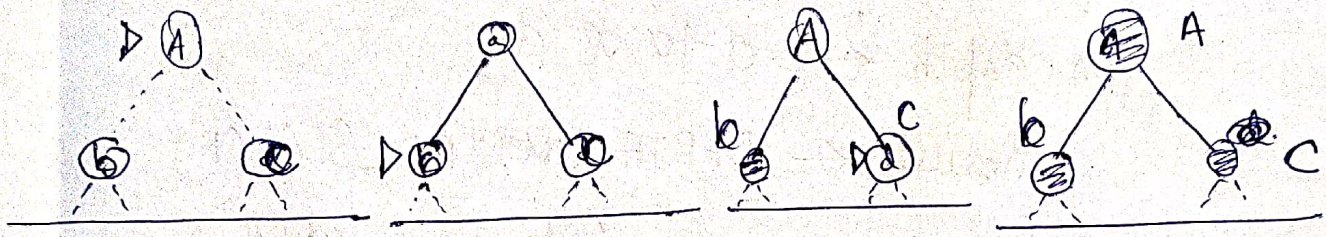
→ If you don't find the Solution then increase depth.

(1) Limit = 0

(Depth - Cutoff = 0)



(2) Limit = 1



Branching factor = b

Depth limit = 1

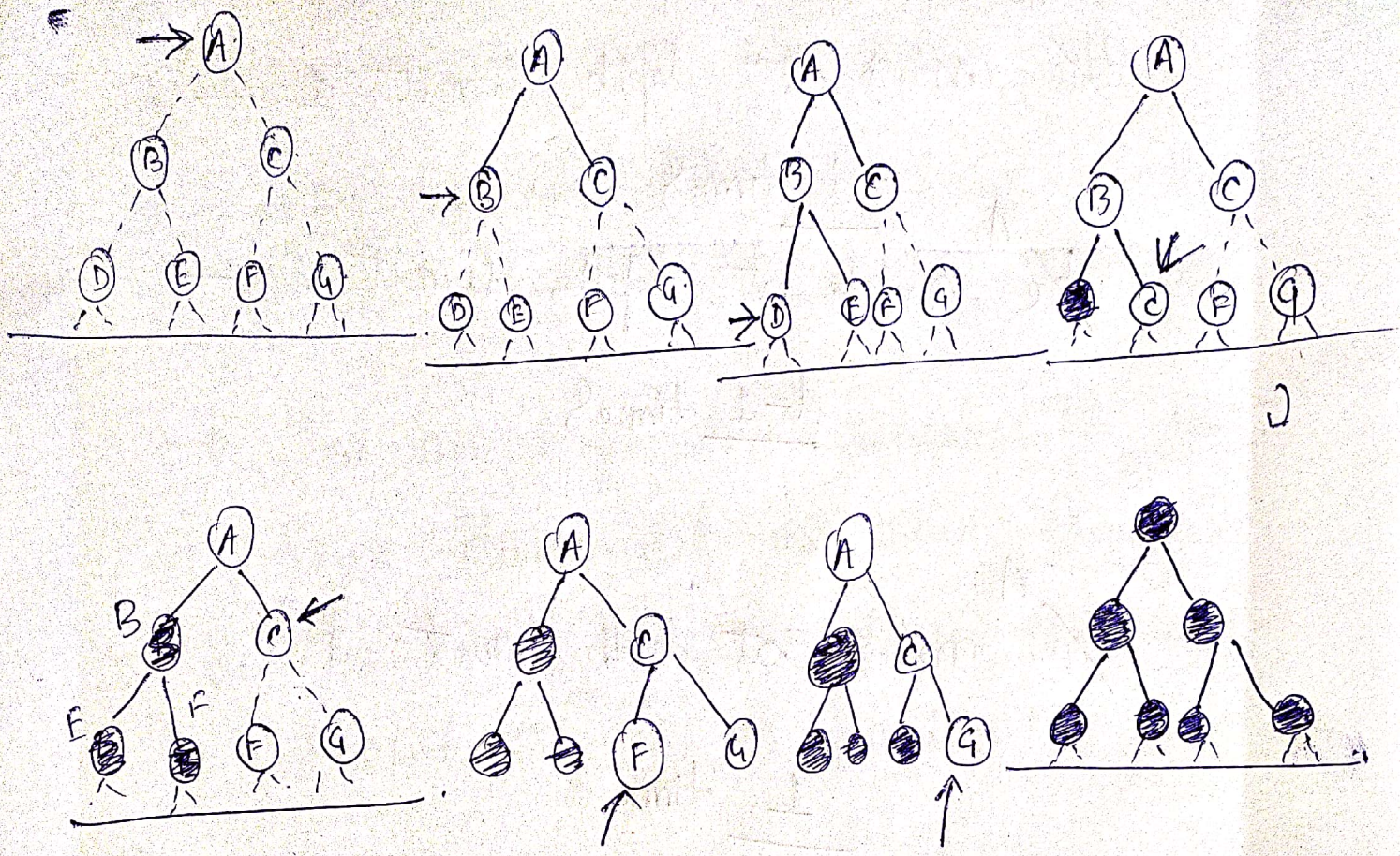
How many nodes are you going to touch.

$b+1$

(3) Limit = 2

No. of nodes you are going to touch

(b^2+b+1)



Analysis of the Algorithm:—

1. Complete : If the goal is at depth 'd', then we are not going to increase the depth.

2. Time :— If the goal is at depth 'd' how many times we are going to touch the top node (root node)

(d+1) times

limit = 0
limit = 1
limit = d.

A
~~the~~ nodes at depth limit 1 :—

d times

A
~~the~~ nodes at depth limit 2 :—

d-1 times.

⋮

A
~~the~~ nodes at depth limit d.

1 time.

$\therefore$ Time $= \underline{(d+1)\,b^0} + \underline{d b^1} + \underline{(d-1)\,b^2} + \cdots + b^d$

$= O(b^d)$ → BFS

AP + GP :

↓ Arithmetic Progression

↘ Geometric Progression.

b^d : Last Set of nodes

Intuition :—

→ Sum up all the nodes $\cong$ to the last level nodes

→ In an Exponential Series the last layer dominates all other layers.
So, the no. of nodes at the last level should as much less as possible.

③ Space :— (DFS)

— $O(bd)$

④ Optimal:

— yes, if Step Cost = 1
— Can be modified to explore uniform Cost Tree.

⑤ Systematic:

(Iterative lengthening)