# Iterative Deepening Method

Active Learning Assignment for the Subject Artificial Intelligence

160410116049 - DEVANGINI KATHAD

160410116050 - KAUSTUBH WADE

160410116051 - NAISARGI KOTHARI

160410116052 - VISMAY LAD

## Why both BFS and DFS are not very efficient?

The problem with DFS is, if there is a node close to root, but not in first few subtrees, then DFS **reaches that node very late.**

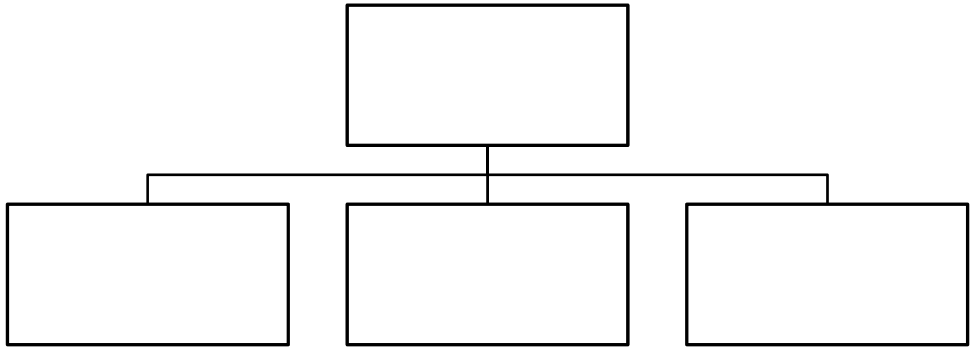
BFS goes level by level, but **requires more space**. The space required by DFS id O(d) where d is depth of tree, but space required by BFS is O(n) where n is number of nodes in tree.

## What is Iterative Deepening?

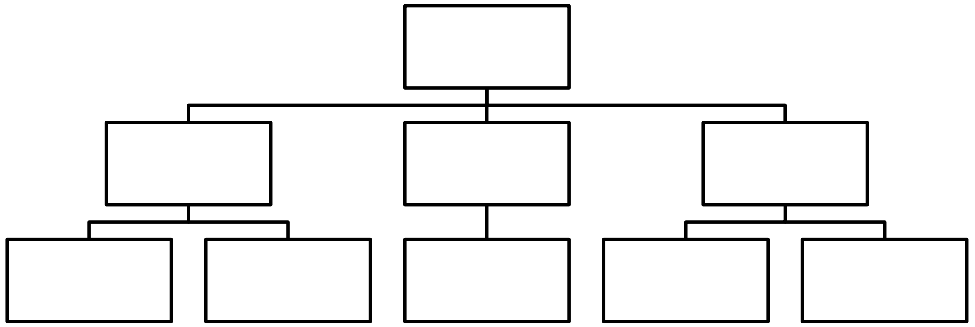
Originally used in a program called CHESS 4.5. Rather than searching to a fixed depth in the game tree, CHESS 4.5 first searched only a single ply, then initiated a new minimax search, this time to a depth of two ply. This was followed by three-ply search, then a four-ply search, etc.

The name “Iterative Deepening” derives from the fact that on each iteration, the tree is searched one level deeper.

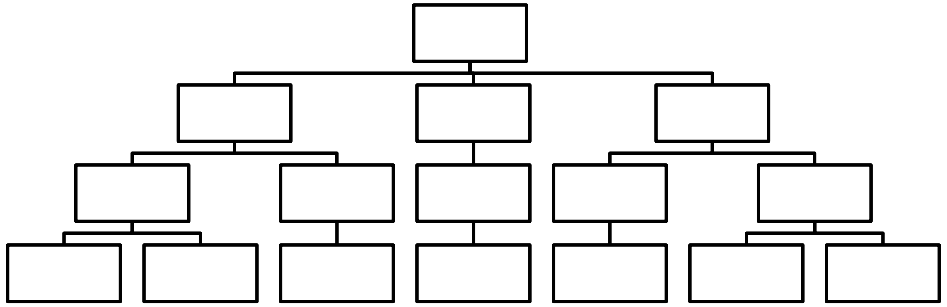
Iteration 1



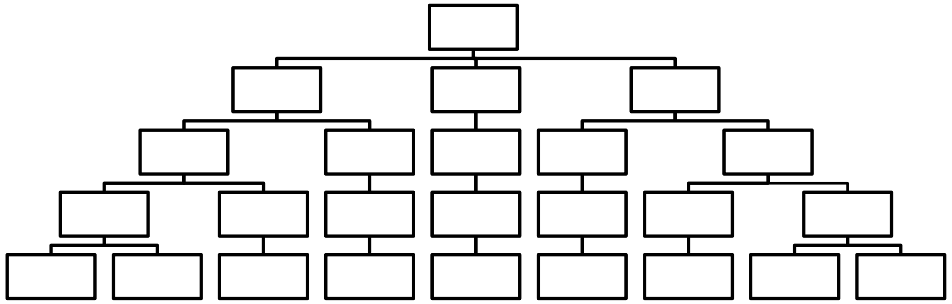
Iteration 2



Iteration 3



Iteration 4



## Why should we use it?

Iterative Deepening calls DFS for different depths starting from an initial value. In every call, DFS is restricted from going beyond given depth. So basically we do DFS in a BFS fashion.

The last (or max depth) level is visited once, second last level is visited twice, and so on. It may seem expensive, but it turns out to be not so costly, since in a tree most of the nodes are in the bottom level. So it does not matter much if the upper levels are visited multiple times.

## Algorithm: Depth-First Iterative Deepening

1. Set SEARCH-DEPTH = 1.
2. Conduct a depth-first search to a depth of SEARCH-DEPTH. If a solution is found, then return it.
3. Otherwise, increment SEARCH-DEPTH by 1 and go to step 2.

## Algorithm: Iterative Deepening-A*

1. Set THRESHOLD = the heuristic evaluation of the start state.
2. Conduct a depth-first search, pruning any branch when its total cost function(g + h’) exceeds THRESHOLD. If a solution path is found during the search, return it.
3. Otherwise, increment THRESHOLD by the minimum amount it was exceeded during the previous step, and then go to step 2.

# Comparison table

| | Time Complexity | Space Complexity | When to use? |
|---|---|---|---|
| DFS | O(bd) | O(d) | Don’t care if the answer is closest to the starting vertex.<br>When graph is not bit. |
| BFS | O(bd) | O(bd) | When space is not an issue.<br>When we do care the closest answer to the root. |
| IDDFS | O(bd) | O(bd) | You want a BFS, you don’t have enough memory, and somewhat slower performance is accepted.<br>In short, when you want a **BFS + DFS** |