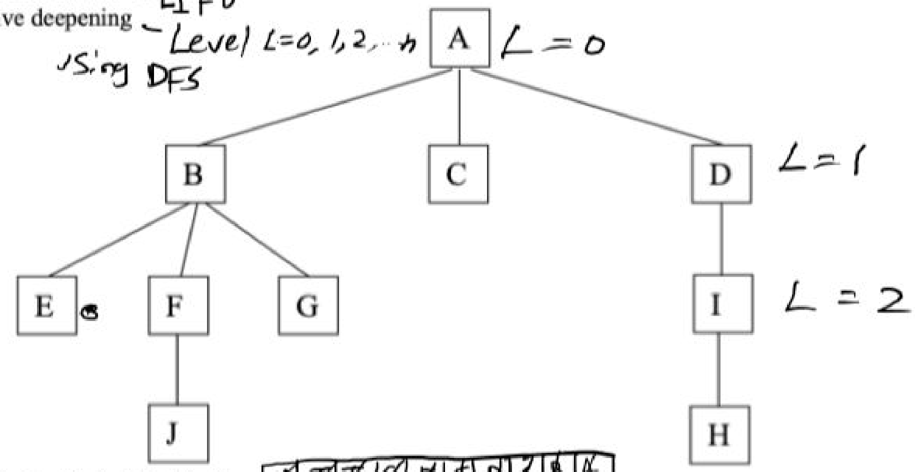
1) In what order would the following algorithms visit the nodes of the search tree below? You can assume that all operators have the same cost. (For clarity, "visit" means *dequeue* from the queuing structure). Hint, for one of these algorithms, we would dequeue 'A" more than once.

A) Breadth first search - level by level (F/FO)
B) Depth first search - LIFO
C) Iterative deepening - Level L=0, 1, 2, ... n
using DFS



A) A, B, C, D, E, F, G, I, J, H $\boxed{\cancel{H}|\cancel{J}|\cancel{I}|\cancel{G}|\cancel{F}|\cancel{E}|\cancel{D}|\cancel{C}|\cancel{B}|\cancel{A}}$

B) $\boxed{\cancel{A}|\cancel{B}|\cancel{E}|\cancel{F}|\cancel{J}|\cancel{G}|\cancel{C}|\cancel{D}|\cancel{I}|\cancel{H}}$ E, J, F, G, B, C, H, I, D, A

C) A, A, B, C, D, A, B, E, F, G, C, D, I, A, B, E, F, J, G, C, D, I, H

2) Suppose we need to optimally solve a problem using blind search, for which the goal node is known to be at a de of *exactly* 17 (i.e. $d = 17$). What algorithm would you use, and why?

Depth-Limited search b/c we know the exact depth.
It's complete. We have knowledge
of depth.

3) Suppose we need to optimally solve a problem using blind search, for which the goal node is known to be at a depth *of no more than* 29 (i.e. *d* <= 29). What algorithm would you use, and why?

Iterative deepening b/c Depth-Limited is

- Complete but not optimal → · Does this mean L is too big So we'll waste time?

yes!

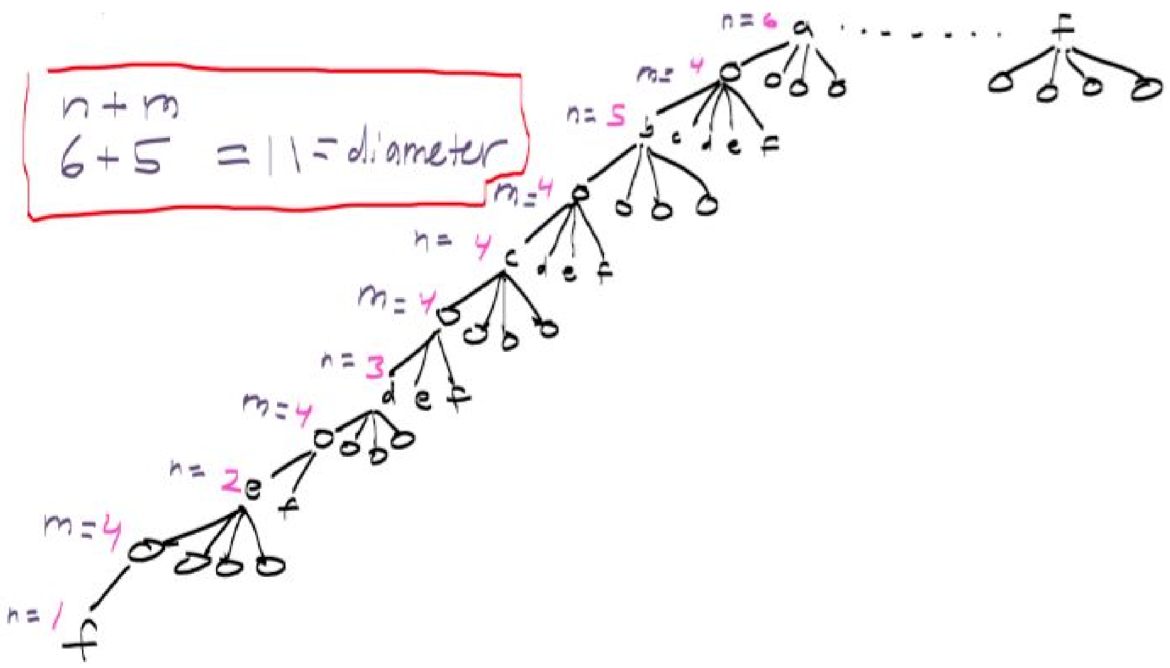
≤ to 29, so Depth would not be optimal, w/ iterative we could find it before 29. Hence, more optimal.

4) What is the *diameter* of a search problem? You could answer with a single English sentence.

The diameter of a problem is the depth of the longest solution or the worst case.

5) What is the diameter of the **countdown numbers game**? (in general, not just the game I show below as an example). Justify your answer with two or three English sentences,
https://www.youtube.com/watch?v=_JQYYz92-Uk&ab_channel=RussellBabidge

5) There are 4 operators (+, −, *, /)
There are 6 numbers a, b, c, d, e, f
We are asked to find the diameter.
No repeats and we do not have to use all the numbers. But since we are finding the diameter we are using all 6 numbers. So we take each number possible and expand each by the # of operators.



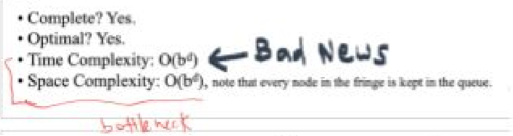
6) Explain the difference between *optimality* and *completeness* for search (one or two sentences)

Completeness is the technique guaranteed to find an answer while optimality is technique used to find the best answer.

7) Suppose we are trying to solve the *Ballyfermot-by-Bike* problem. The branching factor is 17, the only solution is known to be at depth 22. You solve it two ways, Breadth First Search and Iterative Deepening.

- What was the greatest number of nodes in the queue when doing Breadth First Search?
- What was the greatest number of nodes in the queue when doing Iterative Deepening?

7) BFS - with BFS we know that

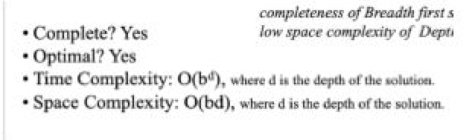


- Complete? Yes.
- Optimal? Yes.
- Time Complexity: $O(b^d)$ ← Bad News
- Space Complexity: $O(b^d)$, note that every node in the fringe is kept in the queue.

bottleneck

- here the branching factor is b=17
- the solution is depth d=22
- It asks for the greatest # of nodes, so here asking for space complexity $\underbrace{O(b^d) = O(17^{22})}$ = 1.17456288×1027 nodes for BFS

Iterative Deepening

- 

  completeness of Breadth first s
  low space complexity of Dept

  - Complete? Yes
  - Optimal? Yes
  - Time Complexity: O(b^d), where d is the depth of the solution.
  - Space Complexity: O(bd), where d is the depth of the solution.

- $O(bd) = O(17 \cdot 22) =$ 374 nodes for iterative deepening.