Name: **Khoa Tran**

Iterative Deepening Depth-First Search (An Example with a Tree)

Let's say we have a problem Π whose problem-space graph can be represented by a tree as shown below. Root node A represents the initial state, and each path descending from the root represents a possible move sequence. Show the order of node visitations by IDDFS, numbering the circles 1, 2, 3, etc., as the search expands each node (removing it from the OPEN list). Each time the IDDFS depth cutoff is increased by 1, move to the next copy of the tree to continue the numbering, back at node A. Assume successors of a node are generated in left-to-right (alphabetical) order.
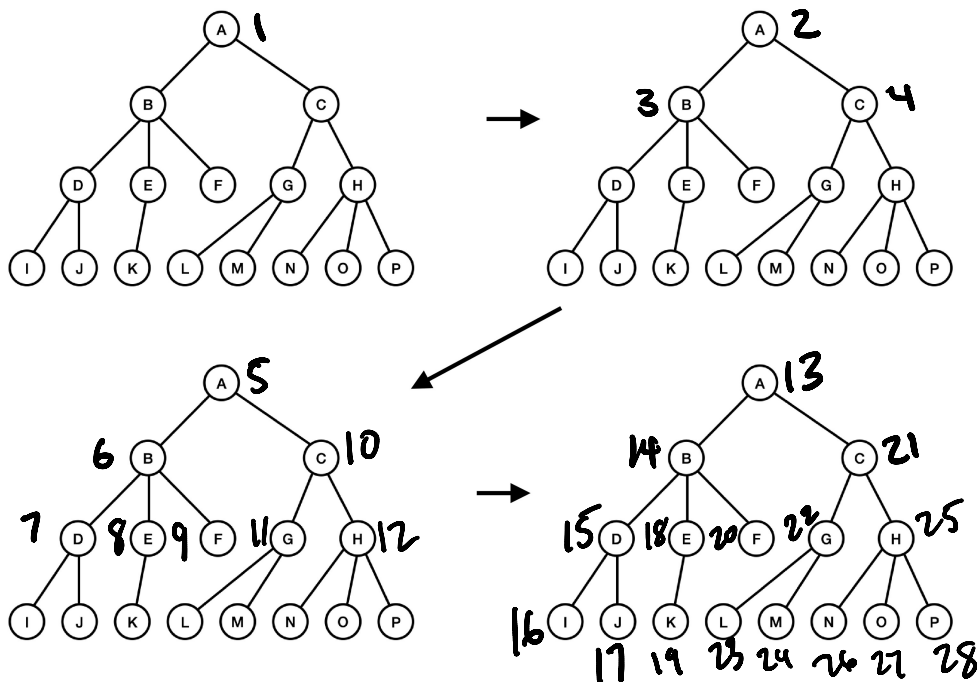
(ii) How many nodes are in the tree? **16**
(iii) How many node visitations are required to reach node G? **11**
(iv) How many node visitations are required to reach node P? **28**
(v) What fraction of (ii) is (iii) ? **11/16**
(vi) What "fraction" of (ii) is (iv) ? **28/16**
(vii) What is the maximum size of the OPEN list when finding G? **4**
(viii) What are the contents of the OPEN list at that maximum size in (vii) ? **(D,E,F,C)**
(ix) What is the maximum size of the OPEN list when finding P? **5**
(x) What are the contents of the OPEN list at that maximum size in (ix) ? **(I,J,E,F,G)**
(xi) Suppose Breadth-First Search is used instead of IDDFS for finding P. What is the maximum size of the OPEN list? **8**
(xii) What are the contents of the OPEN list at that maximum size in (xi) ? **(I,J,K,L,M,N,O,P)**

Breakout mates names and emails: **Steven Hu: sjhu@uw.edu**
Ritik Shah: rhshah@uw.edu
Jin White: jintw@uw.edu