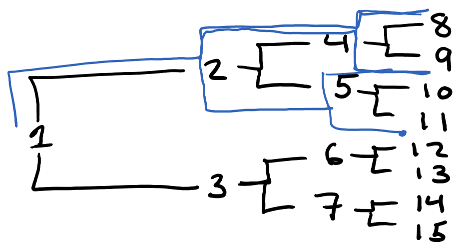
HW #2

Sunday, February 3, 2019 8:18 PM

1. Consider a search problem where the start state is number *1* and the successor function for state *n* returns 2 states, numbered *2n* and *2n+1*. For example, the successors of state 1 are 2*1 = 2 and 2*1+1 = 3, the successors of state 2 are 2*2 = 4 and 2*2+1 = 5, and so forth. In answering the questions that follow, assume the search space has an infinite number of states (violates our prior assumption of finite search problems, but only a mild violation as it is still discrete since the set of states is countably infinite).

   a. Draw a graph of the portion of the search space for states 1 through 15.



2n and 2n+1

· DLS

   b. Suppose the goal state is state 11. List the order in which the nodes will be visited by (assume that the 2n successor is always expanded before the 2n+1 successor):
      ii. breadth-first search
      iii. depth-limited search with depth limit of 3 (Note: depth-limited search is a depth-first search with a limit on how deep into the search tree the search is allowed to go. The root is at depth 0.)
      iv. iterative deepening (begins with a depth limited search with limit 1, if goal not found repeats with limit 2, etc, etc)

ii. BFS

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 Goal

iii. DLS

root 0
DL 3   1 → 2 → 4 → 8 → 9 → 5 → 10 → 11 Goal

iv. ID

DLS Limit 1, 2, 3.....

Depth 1. 1 → 2 → 3

Depth 2. 1 → 2 → 4 → 5 → 3 → 6 → 7

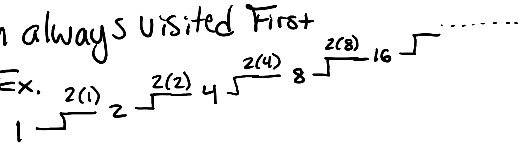
Depth 3. 1 → 2 → 4 → 8 → 9 → 5 → 10 → 11 Goal

   c. Again suppose the goal state is 11, would depth-first search ever terminate (assume that the 2n successor is always visited before the 2n+1)? Note: this question is in reference to the entire search space and not just for the 15 states you drew in part a.

Depth First Search - if the graph has infinite depth the algorithm may not terminate

2n always visited First

Ex.



Since the 2n never will be 11 (the goal) to our Knowledge the DFS will never terminate

d. Again suppose the goal state is 11, would depth-first search ever terminate (assume that the 2n+1 successor is always visited before the 2n)? Note: this question is in reference to the entire search space and not just for the 15 states you drew in part a.

No, DFS will not terminate. Similar to answer above. The graph has infinite depth and 11 (Goal state) is not reached.

2n+1 First

Ex.

$1 \xrightarrow{2(1)+1} 3 \xrightarrow{2(3)+1} 7 \xrightarrow{2(7)+1} 15 \rightarrow \ldots\ldots$

e. Would bidirectional search be appropriate for this problem? If so, describe how it would work. You already have the successors function as stated in the problem. What is the predecessors function for this problem?

Yes it would. BFS would run in a forward and backward direction at the same time.

Forward 2 successors $< \begin{matrix} 2n \\ 2n+1 \end{matrix}$

Backward 1 Successor $\lfloor \frac{n}{2} \rfloor$ ex. $\lfloor \frac{6}{2} \rfloor = 3 \quad \lfloor \frac{5}{2} \rfloor = 2$

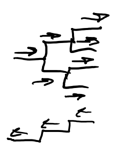
Predecessor function: $\lfloor \frac{n}{2} \rfloor$

f. What is the branching factor in each direction of the bidirectional search?

Kind of answered in (e)

Forward: 2

Backward: 1



Forward 2 successors $< \begin{matrix} 2n \\ 2n+1 \end{matrix}$

Backward 1 Successor $\lfloor \frac{n}{2} \rfloor$ ex. $\lfloor \frac{6}{2} \rfloor = 3 \quad \lfloor \frac{5}{2} \rfloor = 2$

g. Does the answer to part (f) suggest a reformulation of the problem that would allow you to solve the problem of getting from state 1 to a given goal state with almost no search?

Yes, You could Start at the goal and Work Backwards using the 1 Backward Branching Factor and predecessor function $\lfloor \frac{n}{2} \rfloor$.

This would be a lot quicker and require almost no Search.