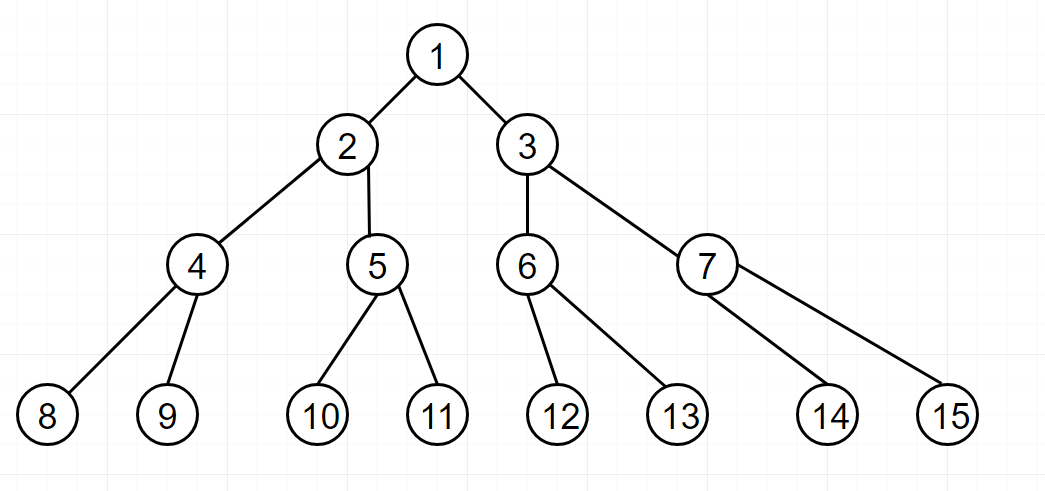
**Chapters 4 & 5 Problem Set**

Your *performance* on this assignment will *not* count towards your homework grade. However, completion of this assignment by the due date *is* required as part of your participation grade. It is primarily for study and preparation for the Midterm (Oct 8).

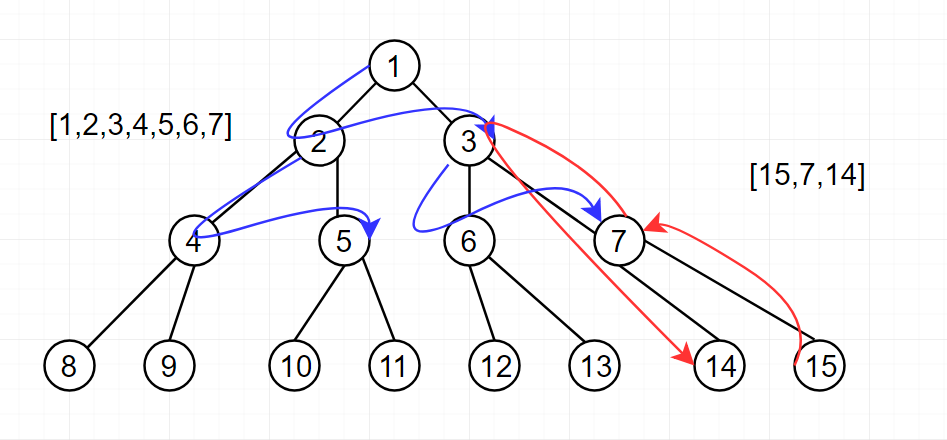
Consider a state space (i.e. search tree) where the start state is number 1 and each state k has two successor states: numbers 2k and 2k+1. Draw the portion of the state space for states 1 to 15.



Suppose that the goal state of the decision tree is Question #1 is 11. List the order in which nodes will be visited for

1. Breadth-first Search (BFS)
   1. [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]
2. Depth-limited Search
   1. [1,2,3,4,5,8,9,10,11,6,7,12,13,14,15]
3. Iterative Deepening Search.
   1. [1,2,4,8,9,5,10,11,6,12,13,7,14,15]

How well would Bidirectional Search work on the problem in Question #2? What is the branching factor in each direction of the bidirectional search?

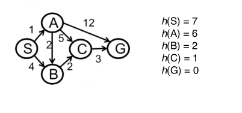


The search finds a solution after processing 10 of the 15 nodes. This is better than the other uniformed methods.

B = 2 in both directions

Does the answer to the search from Question #3 suggest a reformulation of the problem that would allow you to solve the problem of getting from state 1 to a goal state with no (or almost no) search?

Yes, increasing the number of children each node has reduces the amount of searching. When each node has 4 children, a solution can be found after visiting 6 nodes.﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿﻿



Greedy

{source node, dest node} = HueristicCost{dest}﻿﻿

Step1:

-> {S} = 0

Step2:

{S,A} = 6

-> {S,B} = 2

Step3:

->{B,C} = 1

Step4:

-> {C,G} = 0

A\*

{source node, dest node} = PathCost{dest} + HueristicCost{dest}﻿﻿

Step1:

-> {S} = 0

Step2:

{S,A} = 1+6 = 7

-> {S,B} = 4+2=6

Step3:

->{B,C} = 2+1 = 3

Step4:

-> {C,G} = 3+0 = 3

RBFS

{source node, dest node} = PathCost{dest} + HueristicCost{dest}﻿﻿

Step1:

-> {S} = 0

Step2:

{S,A} = 1+6 = 7

-> {S,B} = 4+2=6

Step3:

->{B,C} = 2+1 = 3

Step4:

-> {C,G} = 3+0 = 3