CS141 **Introduction to AI** **Professor Meinolf Sellmann**

Homework 1
Blind Search

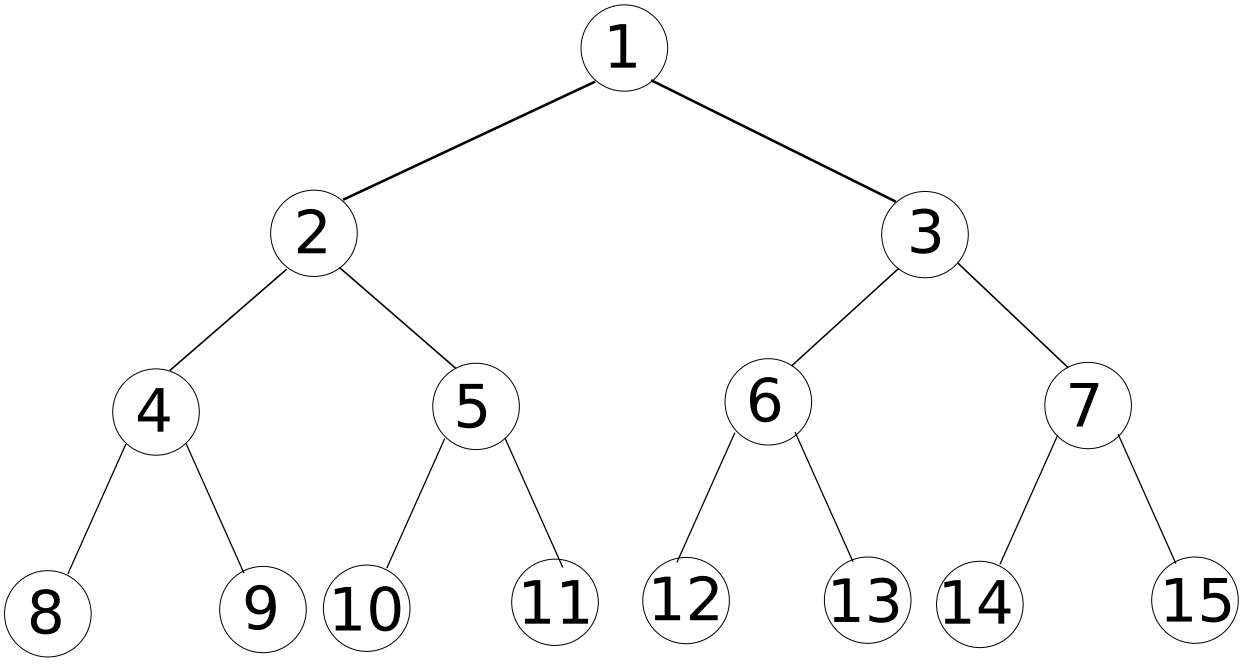
Due: 5:00pm on 2/11/08

Problem 1.1

Do problem 3.8 from the book. When considering iterated deepening, please report those states visited multiple times as such (i.e., *don't* report only the *first* time the node is visited.)

This problem asks you to consider bidirectional search, which often allows for more efficient solutions to search problems by limiting the apparent depth of the search tree. Essentially, instead of just starting at the initial state, you also start at each of the explicitly-defined goal states and walk backwards via a "predecessor function." For more information, check out pages 79 – 80. For Depth-Limited search, take the root node to have depth 1, such that a Depth-Limit 1 search explores the root and nothing else.

Solution: a. see figure.



b. BFS: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

DFS: 1, 2, 4, 8, 9, 5, 10, 11
Depth-Limit 3: 1, 2, 4, 5, 3, 6, 7 (Cutoff; Goal node not reached)
ID: 1; 1, 2, 3; 1, 2, 4, 5, 3, 6, 7; 1, 2, 4, 8, 9, 5, 10, 11

c. Yes, bidirectional search would work well, because there is one explicitly defined goal state and it's easy to generate the predecessor function: if (even) move to $x/2$; else, move to $(x-1)/2$.

d. Branching factor is 2x moving forwards and 1x moving backwards.

e. Search only backwards from the goal state to 1—no branching, and so a very easy problem.

Problem 1.2

Imagine you have the tower of hanoi problem again, but this time there are four pegs on which you can stack disks. Your problem now is to move a stack of four discs from the leftmost peg to the rightmost. Give a shortest series of legal moves such that the pile on the leftmost pile is moved to the rightmost pile. Prove your solution is optimal.

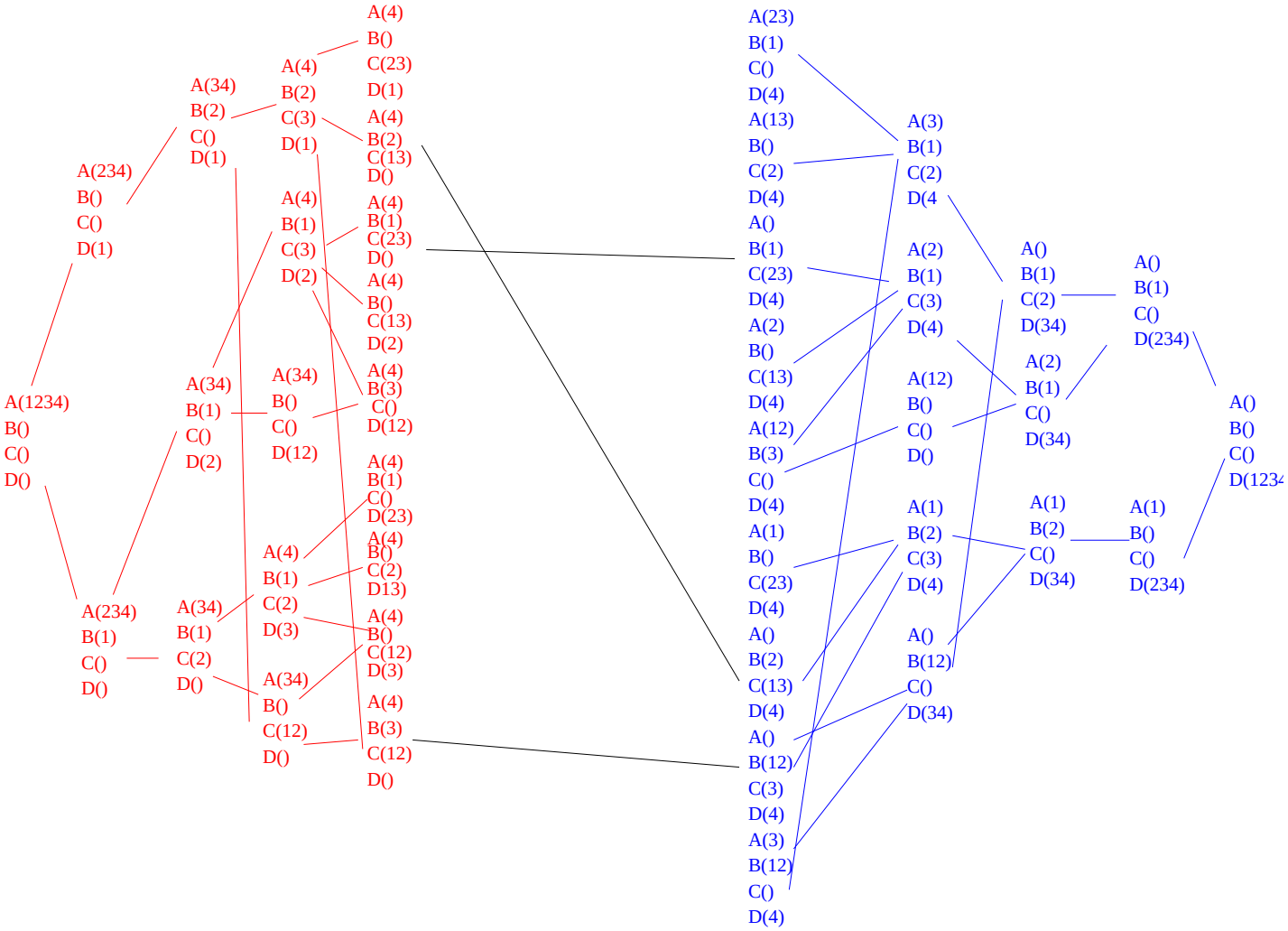
Solution: Name the pegs A, B, C, and D; A is the left peg, D is the right peg. Name the Discs 1, 2, 3, 4; 1 is the smallest disc, 4 is the the biggest disc.

(1) Move Disc 1 from A to B
(2) Move Disc 2 from A to C
(3) Move Disc 1 from B to C
(4) Move Disc 3 from A to B
(5) Move Disc 4 from A to D
(6) Move Disc 3 from B to D
(7) Move Disc 1 from C to A
(8) Move Disc 2 from C to D
(9) Move Disc 1 from A to D

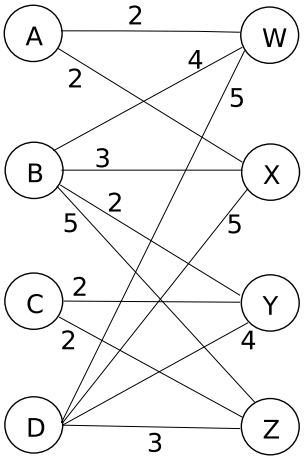
This solution is optimal, but there are other equally fast solutions. One such alternative solution of equal length:

(1) Move Disc 1 from A to B
(2) Move Disc 2 from A to C
(3) Move Disc 1 from B to C
(4) Move Disc 3 from A to B
(5) Move Disc 4 from A to D
(6) Move Disc 3 from B to D
(7) Move Disc 1 from C to B
(8) Move Disc 2 from C to D
(9) Move Disc 1 from B to D

We can show that the solution is optimal through enumerating (intelligently)using bidirectional search. See the attached picture:



Problem 1.3



Sello Construction has 4 backhoes and 4 projects that need backhoes. Backhoes don't get very good gas milage, so Sello Construction has hired you to minimize its gas costs. The goal is to match each of the backhoes A-D with a construction site W-Z. Every backhoe must be matched with exactly one site. Each edge is labeld with a number that represents the cost of using that edge in the matching. You want to minimize the cost of the matching so that Sello Construction saves money and wants to hire you again. What is the optimal matching? How do you know it is optimal? Please show work.

Solution: The minimum matching is A-W, B-X, C-Y, and D-Z, for a total cost of 10. Even if all possible connections from backhoes to projects were allowed, there would only be $4 \times 3 \times 2 \times 1 = 24$ **possibilities. In this case, there are only eight:**

Possible matchings:

A	B	C	D	$TotalCost$
W	X	Y	Z	10
W	X	Z	Y	11
W	Y	X	Z	11
W	Z	Y	X	14
X	W	Y	Z	11
X	W	Z	Y	12
X	Y	W	Z	11
X	Z	W	Y	14

Because we've enumerated all possibilities, we can see that A-W, B-X, C-Y, D-Z is the optimal match.