ICS 271
Fall 2017
Instructor : Kalev Kask
Homework Assignment 1; RN Chapters 3.1-3.4
Due Tuesday, 10/12 in the beginning of class

1. (15 points) The three-disc Tower-of-Hanoi has three pegs A, B, and C, on which can be placed three discs (Labeled $D3$, $D2$, $D1$) with holes in their center to fit the pegs. Disc $D3$ is larger than disc $D2$ which is larger than disc $D1$. We start with the three discs on peg A and want to move them to one of the other pegs except that only the top disc on a peg can be moved, and you cannot place a disc on top of a smaller disc. Let the operators that describe actions be given by the schema $move(x, y, z)$, where x can be any of the three discs $D1$, $D2$, or $D3$, and y and z can be any pair of distinct pegs A, B, or C.

   (a) Define state space for this puzzle,

   (b) Identify the start state and a goal state, and

   (c) Draw the **tree**-search space containing all possible states of the puzzle that are reachable by at most 3 moves. Label the arcs by the appropriate operator. (Each move is reversible; you need label only one of each pair of reversible moves.)

2. (25 points) In the water-jug puzzle, we are given a 5 liter jug, named $Five$, and a 3-liter jug, named $Three$. Initially, $Five$ and $Three$ are empty. Either jug can be filled with water from a tap, T, and we can discard water from either jug down a drain, D. Water may be poured from one jug into the other. There is no additional measuring device. We want to find a set of operations that will leave precisely one liter of water in $Three$. [1]

   (a) Set up a state-space search formulation of the water-jug puzzle:

      i. Give the initial state description as a data structure.

      ii. Define the whole state space.

      iii. Give a goal condition on states as some test on data structures.

      iv. Name the operators on states and give precise descriptions of what each operator does to a state description.

   (b) Draw a graph of all of the distinct state space nodes that are within three moves of the start node, label each node by its state description and show at least one path to each node in the graph-labeling each arc by the name of the appropriate operator. In addition to these nodes, show also all of the nodes and arcs (properly labeled) on a path to the solution.

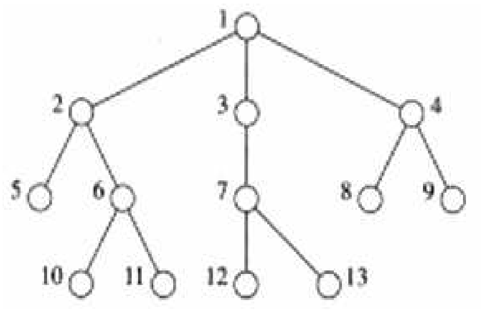
3. (15 points) List the order in which nodes are visited when exploring the state-space given in Figure 1 for each of the following three search strategies (choosing leftmost branches first in all cases):

[1] Here's a solution: (a) fill $Three$ from the tap, (b) pour $Three$ into $Five$, (c) fill Three from the tap,(d) pour as much from $Three$ into $Five$ as will fill it, (e) discard $Five$, (f) pour $Three$ into $Five$.

(a) Uniform Cost Search

(b) Depth-First Search

(c) (Depth-First) Iterative-Deepening Search (increasing the depth by 1 each iteration)



4. (10 points) Assume we are searching a state-space graph where the degree of each vertex is b. However, we do not know if the state-space graph has a tree-like structure or has cycles. So we are considering implementing a graph-search where we check, for each node generated, if the corresponding state has been visited before. Assume that the frontier/explored list of nodes is stored as a hash table with $O(1)$ access time. How many comparisons would have to be made by BFS when expanding all nodes up to and including depth d?

   Extra credit (15 points) : assume the frontier/explored list is stored as a set with $O(n)$ access time. How many comparisons would have to be made by BFS when expanding all nodes up to and including depth d?

5. (15 points) Consider a finite tree of depth d and branching factor b. (A tree consisting of only a root node has depth zero; a tree consisting of a root node and its b successors has depth 1; etc.) Suppose the shallowest goal node is at depth $g \leq d$.

   (a) What is the minimum and maximum number of nodes that might be generated by a breadth-first search?

   (b) What is the minimum and maximum number of nodes that might be generated by a depth-fist search with depth bound equal to d?

   (c) What is the minimum and maximum number of nodes that might be generated by a depth-first iterative-deepening search? (Assume that you start with an initial depth limit of 1 and increment the depth limit by 1 each time no goal is found within the current limit.)