

ICS 271

Fall 2016

Instructor : Kalev Kask

Homework Assignment 1; RN Chapters 3.1-3.4

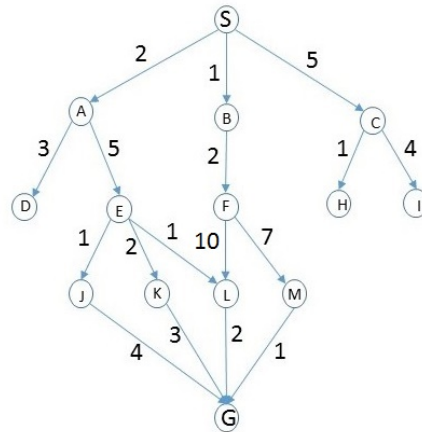
Due Tuesday, 10/11

1. (25 points) In the water-jug puzzle, we are given a 4 liter jug, named *Four*, and a 3-liter jug, named *Three*. Initially, *Four* 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*.<sup>1</sup>
  - (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.
2. (20 points) (Problem 3.9 RN) Three missionaries and three cannibals come to a river. There is a boat on their side of the river that can be used by either one or two persons. How should they use this boat to cross the river in such a way that cannibals never outnumber the missionaries on either side of the river.
  - (a) Specify the state description, the initial state and the goal state for this problem. Describe the state space using variables (as if you are using an array in a program). Determine how many states are in state space.
  - (b) Describe the set of operators using if-then rules.
  - (c) Draw the entire state space graph (include only legal states, that is, states in which cannibals do not outnumber missionaries on either side of the river)
  - (d) Describe a depth-first search algorithm and show a trace leading to a 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):

---

<sup>1</sup>Here's a solution: (a) fill *Four* from the tap, (b) pour as much from *Four* into *Three* without spilling any water, (c) discard *Three*, (f) pour *Four* into *Three*.

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



4. (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-first 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.)
5. (10 points) Assume we are searching a state-space where the degree of each vertex is  $b$ . However, we do not know if the state-space 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  $d$ ?