Summary of Marking Guidelines:

- If you want you can use this as a convenient place to keep your overall instructions to your markers. This way you can find these instructions years later when you are looking back at the exam.

  If you are seeing this and not a cover page, you have solutions set to true in the exam.tex file.

Question 1. [10 MARKS]

This question deals with uninformed search.

Part (a) [4 MARKS]

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 $g \leq d$.

What is the minimum and maximum number of nodes that might be generated by a depth-first search with depth bound equal to d? For full marks show the details of your computation.

Denote by $C(d,b)$ the number of nodes ion a tree of depth d and branching factor b.

$$C(d,b) = \sum_{i=0}^{d} b^i = \frac{b^{d+1}-1}{b-1}$$

In the best case, the first node that a DFS will find at depth g will be the goal. Therefore, the number of nodes generated will be $g+1$ (counting the start node).

In the worst case, we may have to explore all of the tree except the subtree rooted at the goal node. Searching the whole tree generates $C(d,b)$ nodes. The depth of the tree rooted at the goal is $d-g$, and therefore we might have to generate $C(d,b) - C(d-g,b) + 1$ before reaching the goal. (The +1 takes care of adding the goal node back in)

Part (b) [3 MARKS]

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 $g \leq d$.

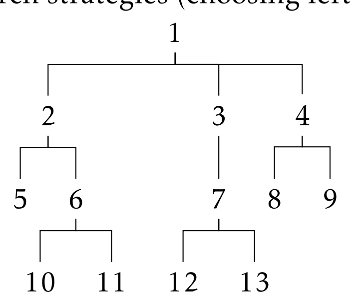
What is the minimum number of nodes that might be generated by a depth-fist iterative-deepening search? (Assume that you start with an initial depth limit of 1 and increment the depth limit by 1 each time no goal is found within the current limit.) For full marks show the details of your computation.

Using same notaion as in the previous part, the best case will happen if the goal were found in the first DFS to level g. In iterative deepening, we first need to search a complete tree of depth 1, then 2, etc. until $g-1$. Therefore the total number of nodes generated would be:

$$g+1+\sum_{i=1}^{g-1} C(i,b) = g+1+\frac{1}{b-1}(\frac{b^{g+1}-1}{b-1} - g)$$

Part (c) [3 MARKS]

List the order in which nodes are visited in the tree in the figure (below) for each of the following three search strategies (choosing leftmost branches first in all cases):



(a) Depth-first search

1,2,5,6,10,11,3,7,12,13,4,8,9.

(b) Depth-first iterative-deepening search (increasing the depth by 1 each iteration)

1,1,2,3,4,1,2,5,6,3,7,4,8,9,1,2,5,6,10,11,3,7,12,13,4,8,9.

(c) Breadth-first search

1,2,3,4,5,6,7,8,9,10,11,12,13.

Question 2. [11 MARKS]

This question deals with A^* path search.

Part (a) [4 MARKS]

Recall tha a heuristic function is monotone if for every node n and its child node n' we have

$$h(n) \leq h(n') + c(n, n').$$

Prove that if h_1, h_2 are both monotone, so is $h = \max(h_1, h_2)$. For full marks please provide all necessary details.

Solution:

$h(n) = \max(h_1, (n), h_2(n))$
$\leq \max(h_1(n') + c(n, n'), h_2(n') + c(n, n'))$
$\leq \max(h_1(n'), h_2(n')) + c(n, n')$
$\leq h(n') + c(n, n')$.

Part (b) [4 MARKS]

The "water jug problem" can be stated as follows: you are given two empty jugs, one that can hold 4 gallons of water and the other that can hold 3 gallons of water. You also have a pump that can be used to fill either jug with water, and you can empty the contents of either jug at any time. Your goal is to get exactly 2 gallons of water into the four-gallon jug.

State this problem in terms of state-space search. Describe the state, the possible actions, initial state, and the goal.

Solution:

- The state is a pair (x, y), where x is the level of the four-gallon jug and y is the level of the three-gallon jug.
- Initial state is $(0, 0)$; goal state is $(2, *)$.
- Successor (x, y):
  - FillX: $(4, *)$
  - FillY: $(*, 3)$
  - EmptyX: $(0, *)$
  - EmptyY: $(*, 0)$
  - PourXY: Let $z = \min(x, 3 - y); (x - z, y + z)$
  - PourYX: Let $z = \min(y, 4 - x); (x + z, y - z)$

Part (c) [3 MARKS]

Suppose that it costs \$5 every time the pump is used, and \$2 every time you use the four-gallon jug, and \$1 every time you use the three-gallon jug. Describe how to find the lowest-cost solution to the problem using A^*. (Keep in mind the word "use" means either fill, or empty, or pour).

Solution

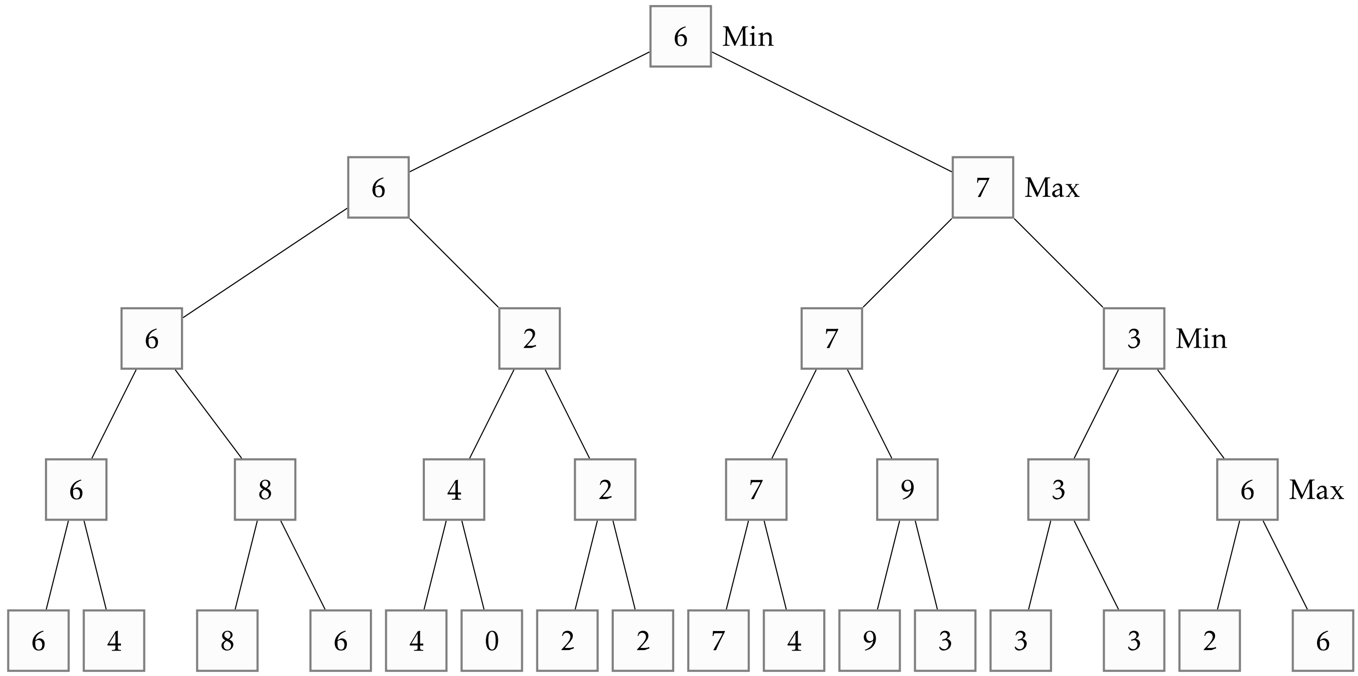
Define $g(x, y)$ function equal to the sum of the action costs. A lower-bound estimate $h(x, y)$ on getting from (x, y) to a goal $(2, *)$:

- if $x = 2, h(x, y) = 0$ for all y.
- if $(x + y) < 2, h(x, y) = 7$ (at least one use of the pump and one fill of 4-gallon jug)
- if $(y > 2), h(x, y) = 3$ for all x (at least one use of both jugs)
- otherwise 1 (at least one use of 3-gallon jug)

Question 3. [6 MARKS]

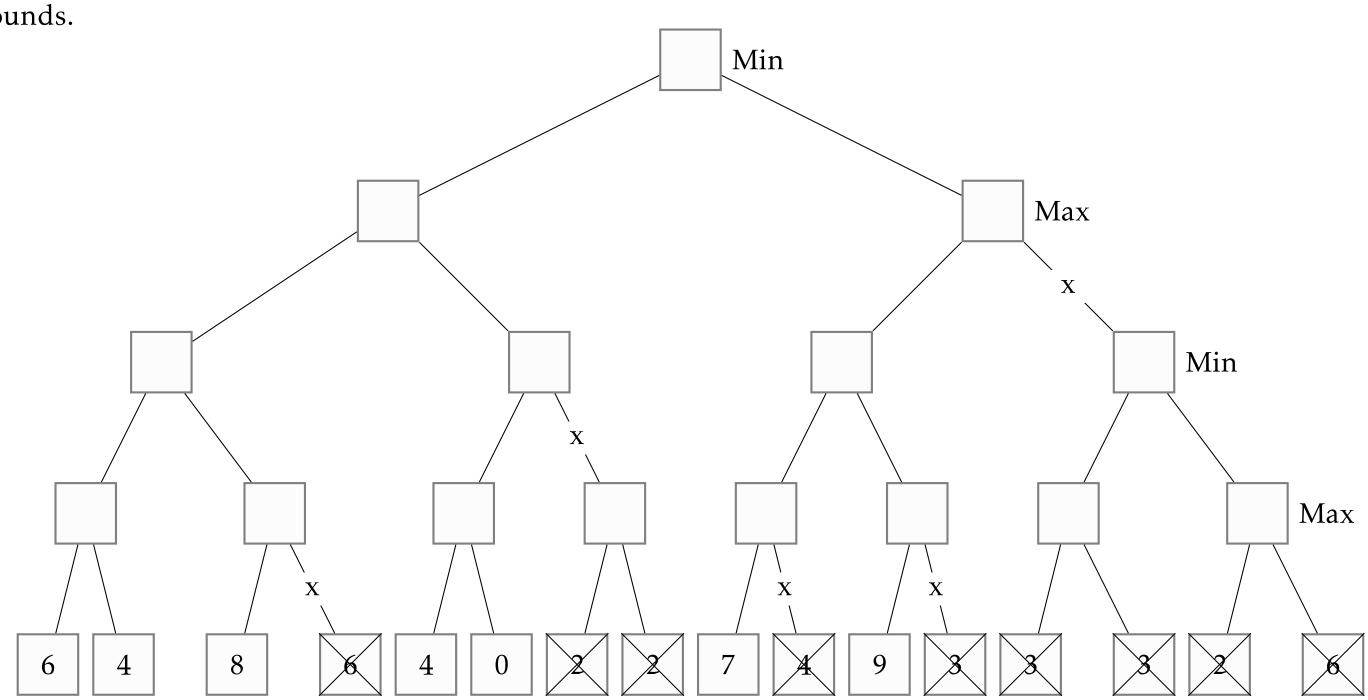
Part (a) [2 MARKS]

The game tree below illustrates a position reached in the game. It is MIN's turn to move. Inside each leaf node is the estimated score of that resulting position returned by the heuristic static evaluator. Fill in each blank square with the proper value according to mini-max search.



Part (b) [4 MARKS]

This is the same tree and conditions as above. Perform alpha-beta pruning. Cross out each pruned branch. Also cross out each leaf node that will not be examined because it is pruned by alpha-beta pruning. You do not need to indicate the branch node values. Also you do not need to indicate the values of alpha/beta bounds.



Question 4. [6 MARKS]

This question deals with CPS. Consider the following problem: there are four variables, X_1, X_2, X_3, X_4 with domains:

$$D_1 = \{1,2,3,4\}, D_2 = \{3,4,5,8,9\}, D_3 = \{2,3,5,6,7,9\}, D_4 = \{3,5,7,8,9\}.$$

The constraints are:

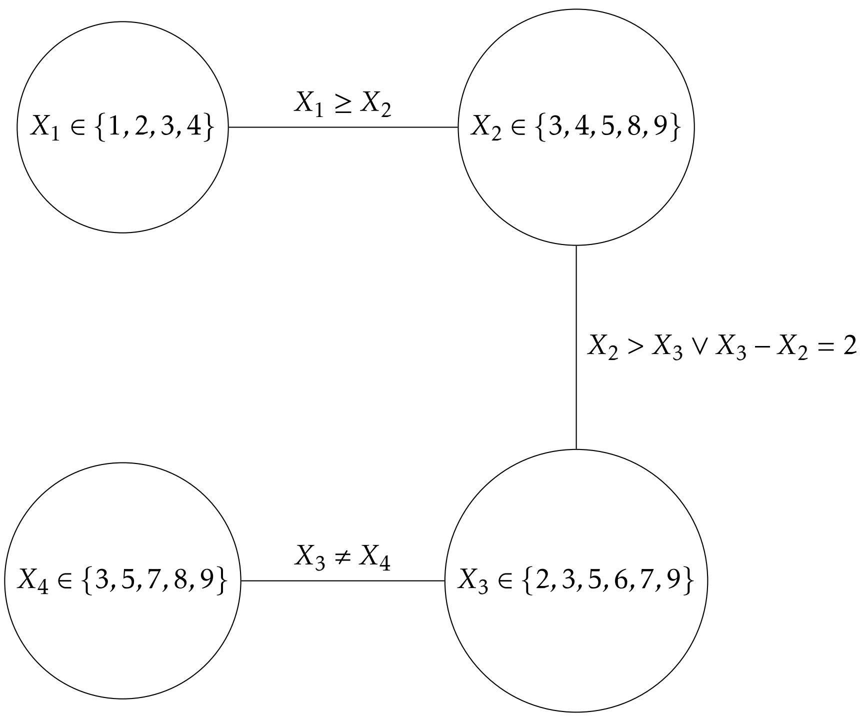
$$C_1(X_1, X_2) : X_1 \geq X_2,$$

$$C_2(X_2, X_3) : X_2 > X_3 \vee X_3 - X_2 = 2,$$

$$C_3(X_3, X_4) : X_3 \neq X_4.$$

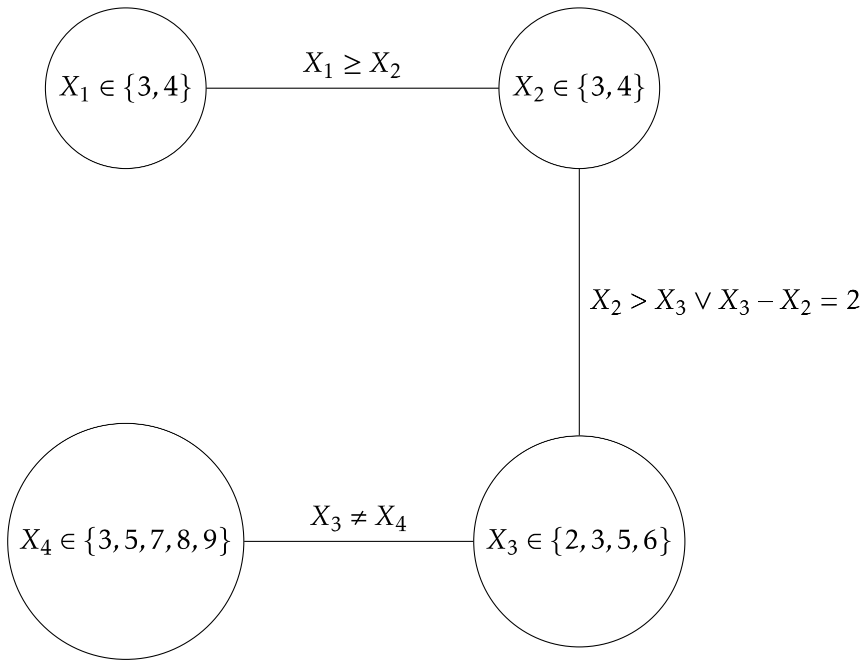
Part (a) [2 MARKS]

Draw the constraint graph. Make sure to show clearly the variables and the constraints as labels on the appropriate elements of the graph.



Part (b) [4 marks]

Is the constraint graph arc-consistent ? If yes, find a solution. If not, compute the arc-consistent graph.



Use the space on this "blank" page for scratch work, or for any solution that did not fit elsewhere.
Clearly label each such solution with the appropriate question and part number.

Total Marks = 33