Assignment 1 – Introductory Concepts

Due date: June 1, 2015 - Midnight

What to submit: a report that contains:

- Matlab/Octave code for the programming exercise (Exercise #1).
- The solutions for the written exercises (Exercises #2,3,4 & 5), typed or neatly handwritten.
- Zip the assignment report and the source code (including a README file) and name it "**Assignment#-Your Project Number#.zip**" such as "**A1-3.zip**"
- Upload this file to **Assignment-1** drop box available on UW LEARN.
- Anything handed in after the due date will be penalized by 50% for each 24 hours of lateness.

I. Graded Exercises

1. **[Programming Exercise – 7 Marks]** Tic-tac-toe (or Noughts and crosses, Xs and Os) is a zero-sum game where players' preferences are opposed, i.e. the winner's reward or payoff is exactly equal to the defeated player's loss. In this game, a single evaluation function is used to describe the goodness of a board with respect to both players. The given code **tictactoe.zip** contains a Matlab implementation for a 3×3 grid. In this implementation, the player who succeeds in placing three respective marks in a horizontal, vertical, or diagonal row wins the game. This playing strategy in this code is a rule-based strategy considering all the possible actions of the opponent and selecting the appropriate player's action accordingly.

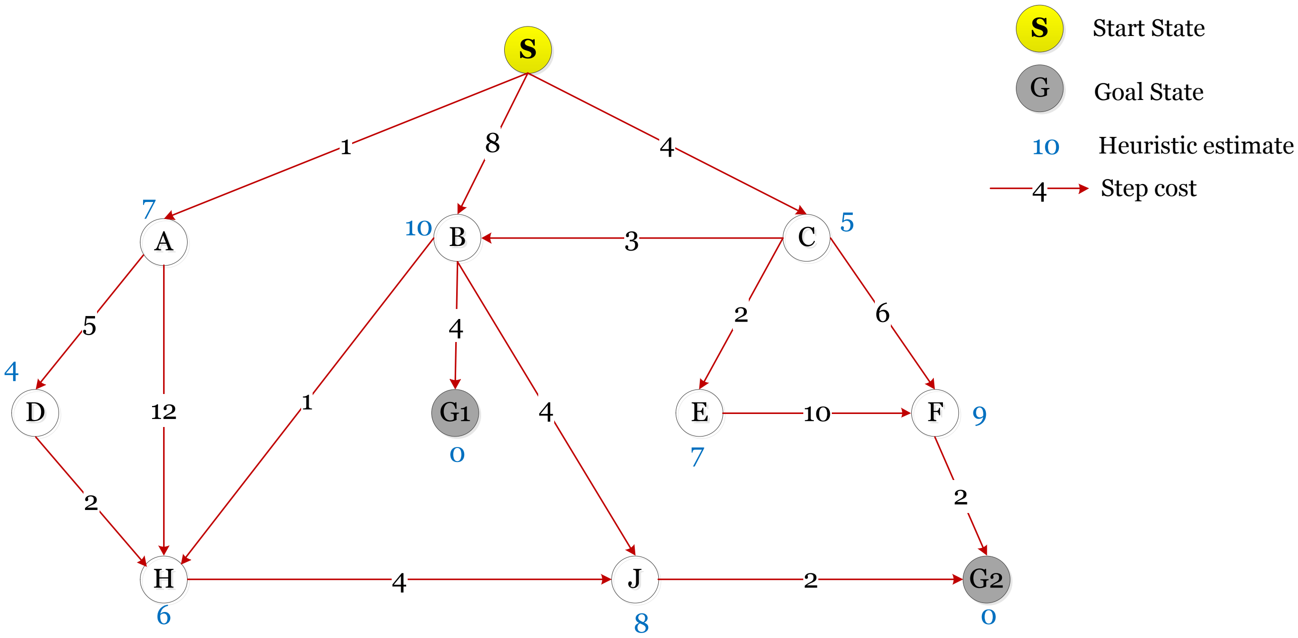
a) Modify the given code by replacing the rule-based strategy by minimax strategy.

b) Modify the implemented minimax strategy by allowing α-β pruning.

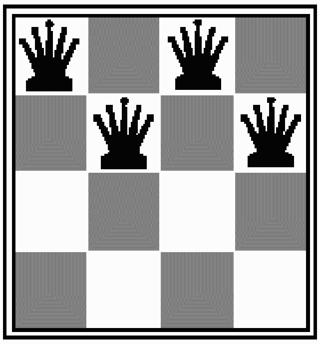
2. **[Written Exercise - 4 Marks]** Consider the search space below, where S is the start node and G1 and G2 are goal nodes. Arcs are labeled with the value of a *cost function*; the number gives the cost of traversing the arc. Above each node is the value of a *heuristic function*; the number gives the estimate of the distance to the goal. Assume that uninformed search algorithms always choose the left branch first when there is a choice. For each of depth-first search (DFS), breadth-first search (BFS), Greedy and A* search strategies,

(a) indicate which goal state is reached first (if any) and

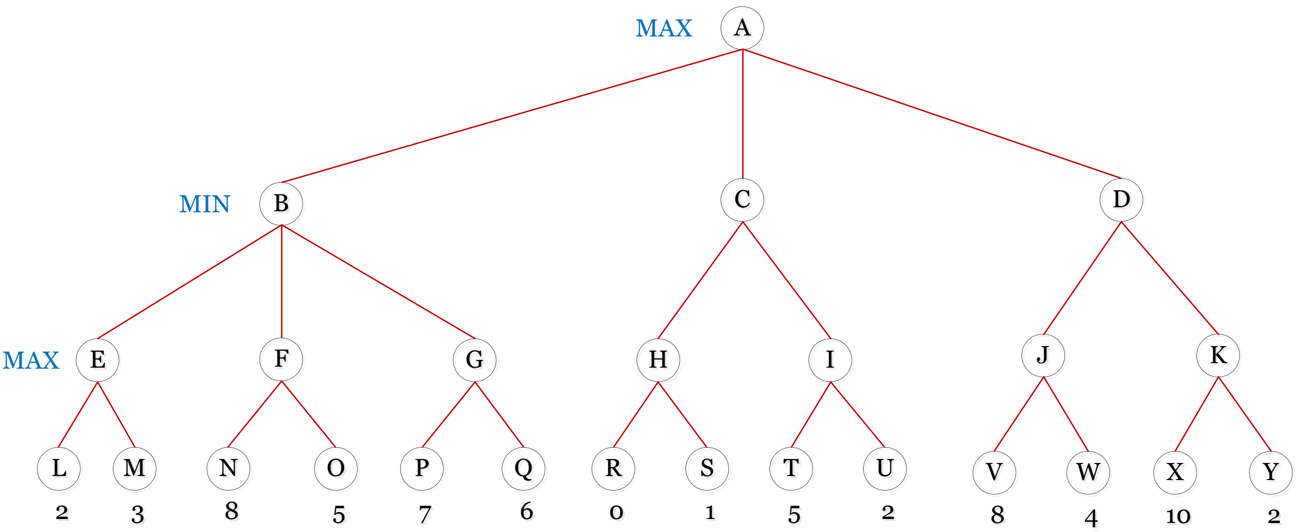
(b) list *in order*, all the states that are popped off the OPEN list.



3. **[Written Exercise - 2 Marks]** N-queen problem is an example for constraint-satisfaction problem that does not define an explicit objective function. Instead, the objective is to find a solution which satisfies all of a set of constraints. In N-queen problem, the goal is to put *n* queens on an *nxn* board with no two queens on the same row, column, or diagonal. Starting from the given configuration, move the queens to reduce number of conflicts until you reach the final configuration where no queen is attacking another queen.



4. **[Written Exercise - 2 Marks]** Consider the following game tree in which the root corresponds to a MAX node and the values of a static evaluation function, if applied, are given at the leaves.



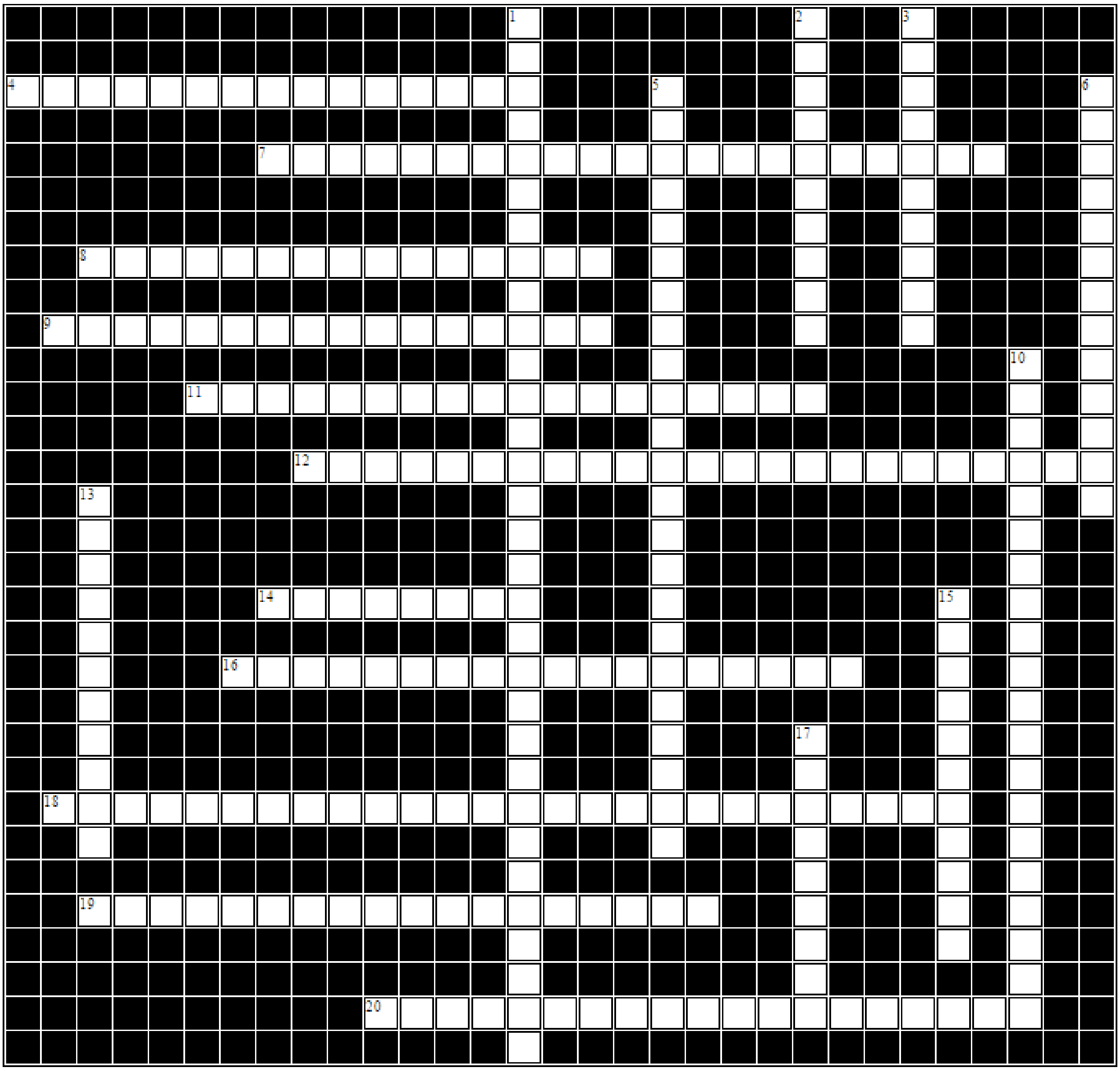
a) What is the minimax value computed at the root node for this tree?

b) What move should MAX choose?

c) Which nodes are not examined when α-β pruning is performed? Assume children are visited left to right.

d) Is there a different ordering for the children of the root for which more pruning would result by α-β? If so, state the order. If not, say why not.

5. **[Written Exercise - 5 Marks]** Solve the following crossword puzzle



Across

4. a form of exploration to cause the search to consider new areas. Examines unvisited regions, generates different solutions.

7. a feasible solution that provides a superior objective function value, but not necessarily the best.

Down

1. a problem that does not define an explicit objective function. Instead, the objective is to find a solution which satisfies a set of constraints.

2. a guessing game for two players where not all state information is available to

8. a problem constraint that is desirable to satisfy.

9. algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding the global optimum.

11. a graph search algorithm for traversing or searching tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking.

12. a problem that can be stated in terms of numerical variables. Its goals can be specified in terms of a well-defined objective function and there exists an algorithmic solution for it.

14. a graph search algorithm that solves the single-source shortest path problem for a fully connected graph with nonnegative edge path costs, producing a shortest path tree.

16. a procedure which can prune large parts of the search tree and allows search to go deeper.

18. a part of computer science devoted to solution of non-algoritmizable problems.

19. any algorithm able to adjust to new or different situations or to improve behaviour or evolves and learns from instructor, example or by discovery.

20. an NP-hard problem.

all players and outcome of actions is deterministic.

3. a solution strategy or rules by trial-and-error to produce acceptable (optimal or near-optimal) solutions to a complex problem in a reasonably practical time.

5. a problem that specifies more than one sub-objective which need to be simultaneously optimized.

6. is a game where the players' preferences are opposed. The winner's reward or payoff is exactly equal to the defeated player's loss.

10. an optimization technique in which time is not so important and a user is willing to wait maybe even days if he/she can get an optimal or close-to-optimal result.

13. an ordered combination.

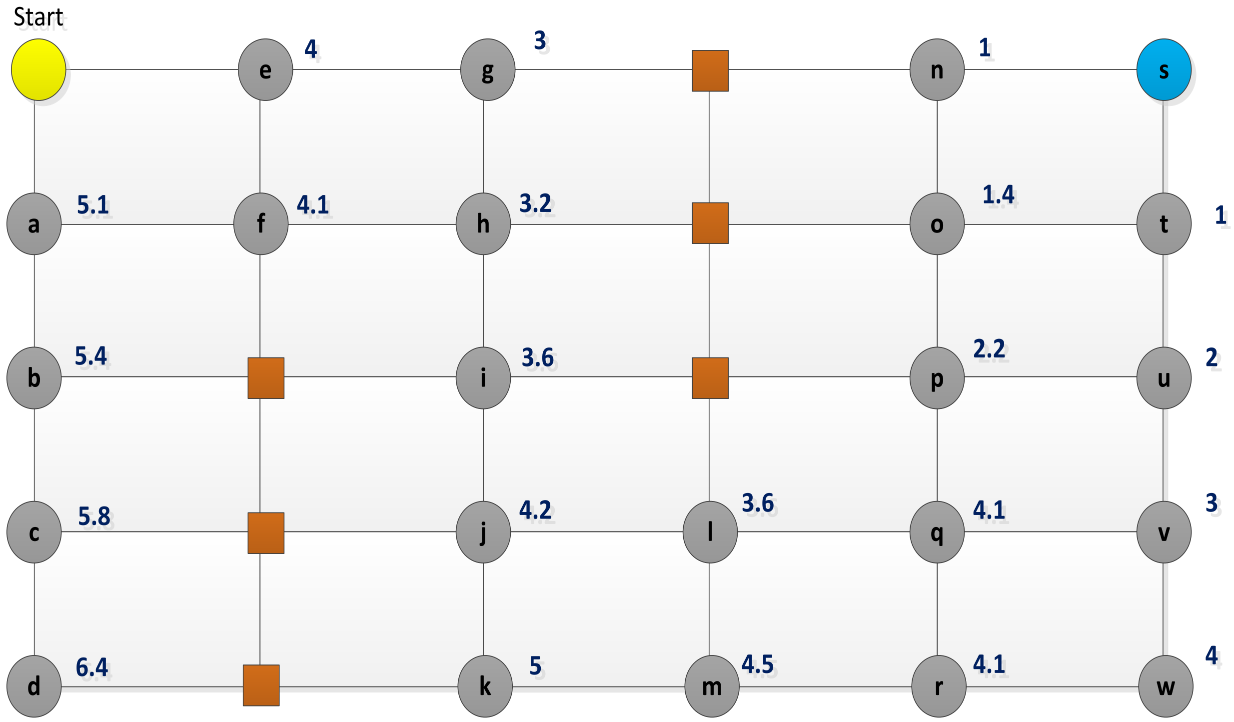
15. an informed breadth-first algorithm.

17. Asymptotic notation used to describe asymptotically tight bounds of a computer algorithm.

Hint: Spaces and dashes MUST be used if the answer consists of two or more words.

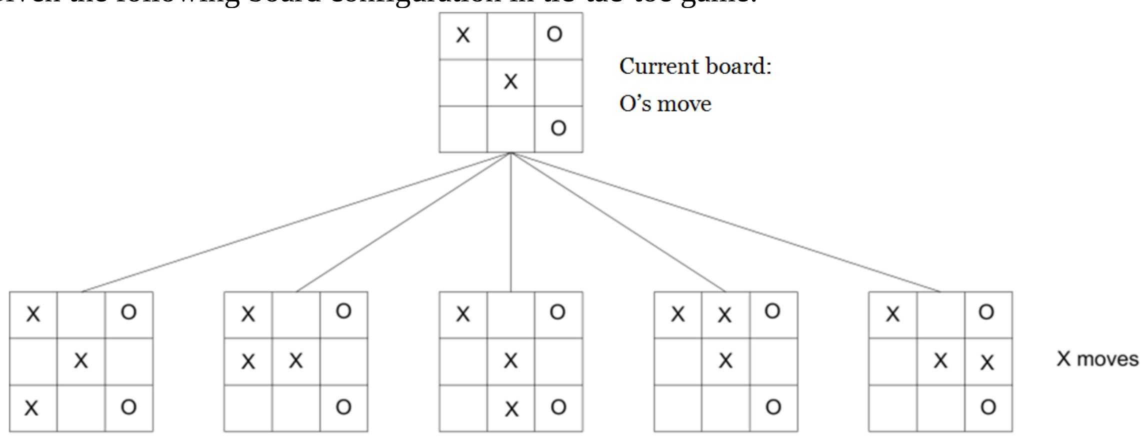
II. Non-Graded Extra Exercises

1. Path Planning is considered as one of the core problems of autonomous mobile robots. This problem is an interesting problem that has been studied extensively over the last two decades. Different approaches have been proposed with different levels of complexity, accuracy and applicability. This problem addresses how to find a collision-free path for a mobile robot from a start position to a given goal position, amidst a collection of obstacles. Path planning can be seen as an optimization problem which aims at improving some system objectives with several constraints. System objectives may be the travelling distance, travelling time, and consuming energy. However, the travelling distance is the most common objective. Consider planning an optimal path for a mobile robot to move from Start position to Goal position S in the following 4x5 grid environment. In this environment, the red squares represent obstructive positions.



Show how to use A* to find the shortest path between the start position “Start” and the goal “s”. Assume that the heuristic value highlighted in blue is calculated using the Euclidian distance, which represents the distance from a point to the goal.

2. Given the following board configuration in tic-tac-toe game:



Assume that *f*(n) is evaluation function or static evaluator used to describe the goodness of a board with respect to MAX player. *f*(n) = [# of 3-lengths open for the MAX player] - [# of 3-lengths open for MIN player or opponent]. Calculate the f(n) for each of the X moves shown in the above board.

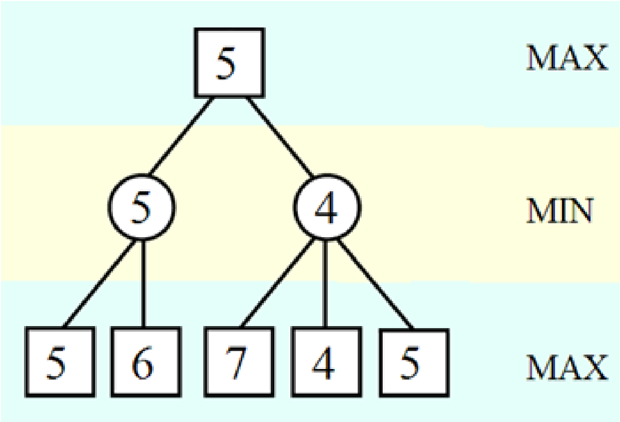
3. MCQs & T/F.
   a) What is the type of this optimization problem: max x+y subject to 0 < x < 5, 0 < y < 2 and 2x + y = 8.
      - Unconstrained multiobjective optimization
      - Multiobjective optimization with multiple constraints.

- Constrained nonlinear optimization with a single objective (uni-objective).
- None of the above

b) For very large workspace where the goal is deep within the workspace, the number of nodes could expand exponentially and depth-first search will demand a very large memory requirement.

- True
- False

c) For the following game tree, which nodes will be pruned? Assume children are visited left to right.



- Node 6 (child of 5) in the MAX layer
- Node 7 (child of 4) in the MAX layer
- Node 4 (child of 4) in the MAX layer
- Node 5 (child of 4) in the MAX layer
- All of the above
- None of the above