CS 480 Artificial Intelligence Fall 2009

Take-home Final Examination, Due: Friday 10 AM

This is an open-book, open-notes exam. Access to any online material is also permitted, but no discussion with any one else while the exam is in progress. If you are not clear about a question, please send me an e-mail and I will try to respond as quickly as possible. Alternatively, you can make reasonable assumptions, state them explicitly and proceed.
For the programming problem, please include the screen shot of the outputs for the specified inputs.

1) Consider the following problem: we start with an array containing an arbitrary permutation of the integers 1, 2, 3, …, N. The goal is to sort this array (i.e., to transform it into the identity permutation) using the fewest number of operations. The only operation allowed is to select a contiguous block (i.e., subarray a[i .. j] for some i < j) and reserve it. For example, (with N = 20), from the permutation [1 3 5 7 … 19 2 4 6 8 10 … 20], one can obtain the permutation [1 3 5 7 … 19 20 18 16 … 4 2] by reversing the block containing the last 10 elements.

   (a) Formulate this problem as a search problem, by defining the state space, the successor function, the initial state, and the goal state.

   (b) How many nodes are in the state space (as a function of N)? How many outgoing edges are there from each node?

   (c) Is the search graph cyclic? (i.e., does it have cycles of length 3 or more?)

   (d) Write a predicate **move(X, Y)** in Prolog that returns true (false) if and only if the permutation Y can be obtained from X in a single move. For example, **move([1 3 4 2 5 6], [1 3 5 2 4 6])** should return true while **move([1 3 4 2 5 6], [1 2 3 4 5 6])** should return false. Your program need not check that X and Y are permutations.

2) Consider the search problem illustrated by the graph of Figure 1. Each node of the graph represents a state. So, the state space is {S, A, B, C, D, E, G}. The initial state is S and the goal state is G. Each arc represents a possible transition from one state to another and the number besides each arch is the cost of the transition. We would like to use the A* algorithm to generate a minimum-cost path from S to G, where the cost of a path is the

sum of the costs of the arcs forming this path. The h values given for all states are the values of a heuristic function to be used by A*.

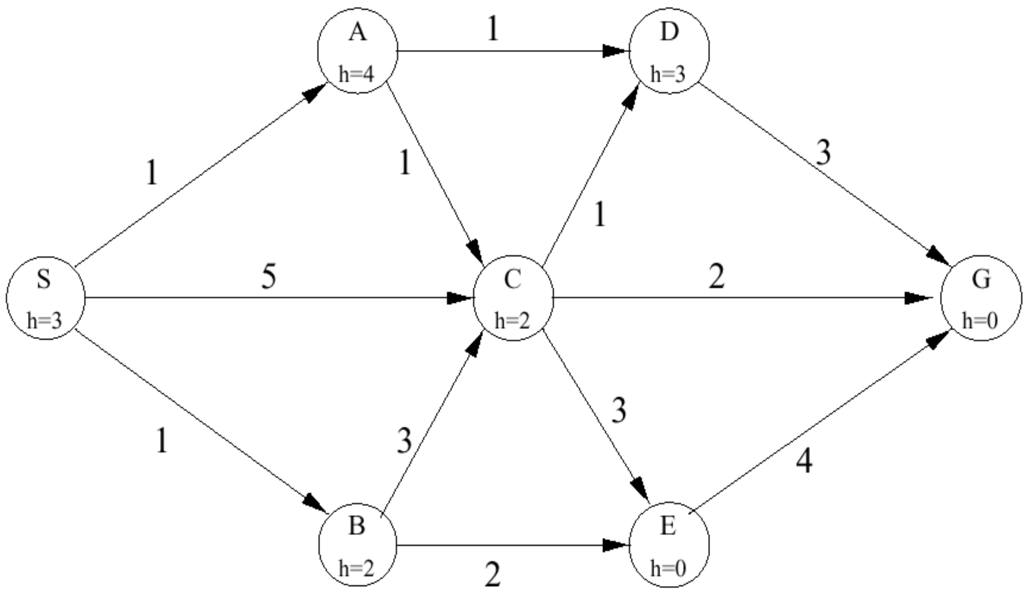

Figure 1

(a) Exhibit the sequence of nodes visited by iterative deepening search (IDA) algorithm.

(b) Write down the sequence of nodes expanded by A* algorithm.

(c) What is the optimal cost of the solution?

(d) Which of the following algorithms finds the optimal solution? DFS, IDA, A*

3. (a) How many models does the proposition $(A \wedge B) \Rightarrow (B \wedge C)$ have?

(b) State if each of the following sentences is valid, unsatisfiable, or neither:

SMOKE $\Rightarrow$ FIRE
(SMOKE $\Rightarrow$ FIRE) $\Rightarrow$ ($\neg$FIRE $\Rightarrow$ $\neg$SMOKE)
(SMOKE $\wedge$ (SMOKE $\Rightarrow$ FIRE)) v $\neg$FIRE

Explain your answers.

(c) Consider the following sentences in propositional logic:

(1) Battery-OK $\wedge$ Bulbs-OK $\Rightarrow$ Headlights-Work
(2) Battery-OK $\wedge$ Starter-OK $\wedge$ $\neg$Empty-Gas-Tank $\Rightarrow$ Engine-Starts
(3) Engine-Starts $\wedge$ $\neg$Flat-Tire $\Rightarrow$ Car-OK
(4) Radio-Works $\Rightarrow$ Battery-OK
(5) $\neg$Headlights-Work
(6) $\neg$Empty-Gas-Tank
(7) Radio-Works
(8) Starter-OK
(9) $\neg$Flat-Tire

Transform these sentences into clauses.

(d) Use resolution to prove Car-OK. Present your proof with one clause per line. Label each clause by an integer (use consecutive integers) and indicate after each derived clause the numbers of their parents.

(e) Exhibit a proof of the assertion Car-OK using forward chaining.

4) Consider a coin weighing problem similar to the one we discussed in class: there are 7 coins of which two are odd coins - one is heavier than the rest and the other is lighter the rest by the same amount. (Thus the total weight of the two odd coins = the total weight of two normal coins.) The goal is to identify the odd coins. All coins look identical but a balance can be used to weigh some coins against others. Consider the three options:

Option 1: weigh 1, 2 against 3, 4
Option 2: weigh 1, 2, 3 against 4, 5, 6
Option 3: weigh 1 against 2

In each case, determine the information gain (or equivalently the entropy reduction). Which option is the best in terms of the information gain?

5) (a) Let N(f) and N(g) be two neural networks for two Boolean functions F and G (both of n Boolean variables $x_1, x_2, \ldots, x_n$). The neural network output is 1 (representing true) or 0 (representing false) according as $F(x_1, x_2, \ldots, x_n)$ = True or False and similarly for G. Describe a neural network for the Boolean function F V ~G by combining N(f) and N(g) with one extra neuron.

(b) Let N(f) and N(g) be two neural networks for Boolean functions F and G as above. Describe a neural network for the Boolean function $F \oplus G$ by combining N(f) and N(g) and adding as few additional neurons as possible.

(c) If a (finite) data set representing a Boolean function f of three real variables is linearly separable, is it true that the projection of f on XY plane, i.e., $f_x(y,z) = f(x,y,0)$ linearly separable? If the answer is YES, prove it. If the answer is NO, give a counter-example.

(d) Show that if a Boolean function f of three real variables is linearly separable, then there exists a plane on which the data can be projected so that the projected points are linearly separable. (It is enough to describe the plane on which to project the data. You need not formally prove that your construction works.)

6) Consider the following game tree for a two-player game. The tree is represented as a list in which A is the root with 7 children B, F, G, … , K. The list (B (E (L[2] M[5])) represents the subtree rooted at B, etc. The static scores indicated between brackets besides the tree leaves are from the first player's point of view:

```
(A   (B  (E   (L[2]  M[5]))
     (F  (N[3] O[4]))
     (G  (P[4] Q[1])))
     (C  (H (R[-1] S[0]))
     (I  (T[7] U[3])))
     (D  (J (V[2] W[3]))
     (K  (X[4] Y[2]))))
```

The first player is the maximizing player MAX. Perform depth-first search (with the ordering of the nodes as listed above) of the game tree with $\alpha-\beta$ pruning. Give the final α or β value at each node of the game tree that has not been pruned and indicate all the nodes for which no α or β value has been computed (i.e., indicate the pruned nodes). Based on this tree search, what is the optimal move determined by the first player?

7) Given below is a training set of data containing various attributes, along with a classification (rich or not):

Name	Area	Resources	GNP/person	population	Rich?
Canada	Big	Yes	High	30	Y
Russia	Big	Yes	Med	150	N
Brazil	Big	Yes	Low	160	N
Switzerland	Small	No	High	20	Y
China	Big	Yes	Low	1300	N
UK	Med	No	High	55	Y
Spain	Med	No	High	40	Y
Australia	Big	Yes	High	20	Y
Egypt	Med	No	Low	60	N

Build a naïve Bayes classifier for the above data and use it to classify country X (as Rich or Not Rich) based on the following attributes:

Country X	Med	Yes	High	300

(Assume that population forms a normal distribution.)