15-381 Spring 05 Midterm

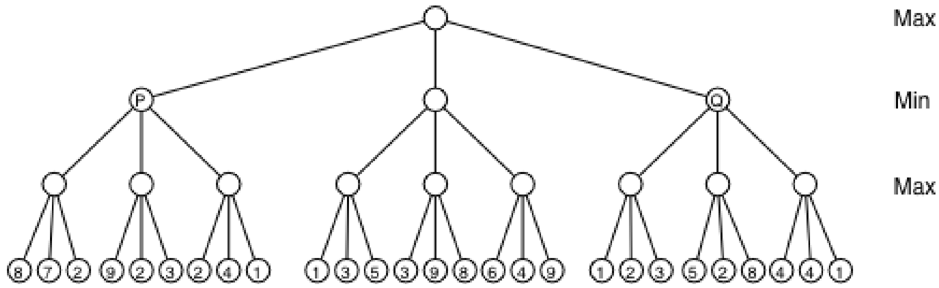
Tuesday March 1, 2005

Name: ______________________________ Andrew ID: ______________________________

- This is an open-book, open-notes examination.You have 80 minutes to complete this examination.
- This examination consists of 6 questions, each worth 20 points. For each student, **only the top 5 scoring questions** will be considered. The worst-scoring question will be discarded (thus you may choose to ignore a question and still potentially get full marks). The maximum possible score is 100.
- Write your answers legibly *in the space provided* on the examination sheet. If you use the back of a sheet, indicate clearly that you have done so on the front.
- Write your name and Andrew ID on this page and your andrew id on the top of each successive page in the space provided.
- Calculators are allowed but laptops and PDAs are not allowed.
- Good luck!

1 Problem 1: Game Tree Search (20 pts)

The figure below is the game tree of a two-player game; the first player is the maximizer and the second player is the minimizer. Use the tree to answer the following questions:



(a) What is the final value of this game?

Consider running the alpha-beta pruning algorithm on this game tree.

(b) Is the final value of beta at the root node (after all children have been visited) $+\infty$? (**T/F**)

(c) What is the final value of beta at the node labeled P (after all of P's children have been visited)?

Suppose we are in the middle of running the algorithm. The algorithm has just reached the node labeled Q. The value of alpha is 5 and the value of beta is $+\infty$.

(d) Will any nodes be pruned?

(e) What value will Q return to its parent?

2 Problem 2: Game Theory (20 pts)

Two players, A and B, play a game, in which they each shout out an integer: 1, 2 or 3. If they both shout the same number, they receive a prize:

- They each get one dollar if they both shouted "1".
- They each get two dollars if they both shouted "2".
- They each get three dollars if they both shouted "3".

If they shouted different numbers, they get nothing.

(a) Is it a Nash Equilibrium to both shout "1"? (**T/F**)

Consider the mixed strategy of

I1: shout "1" with probability $\frac{1}{3}$
I2: shout "2" with probability $\frac{1}{3}$
I3: shout "3" with probability $\frac{1}{3}$

(b) If both players use this mixed strategy, is that a Nash Equilibrium? (**T/F**)

Now consider a very different game. Two companies, A and B, both make elbow warmers. The more they spend on advertising, the more sales they get, but there are diminishing returns. A's advertising somewhat helps B, and B's advertising somewhat helps A. the exact fomulas are

$$\begin{aligned} P_a &= \overbrace{\log(2a+b)}^{revenue} - \overbrace{a}^{expense} \\ P_b &= \log(a+2b) - b \end{aligned}$$

where

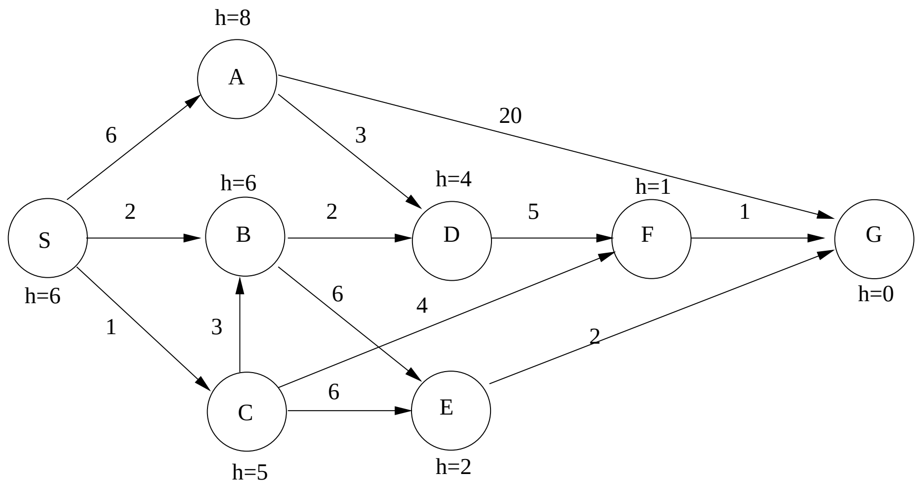
$$\begin{aligned} \text{a} &= \text{\# dollars that A spends on advertising} \\ \text{b} &= \text{\# dollars that B spends on advertising} \\ P_a &= \text{profit to A} \\ P_b &= \text{profit to B} \end{aligned}$$

(c) Work out the nash equilibrium.

Hint: $\frac{\partial P_a}{\partial a} = \frac{2}{2a+b} - 1 \quad \frac{\partial P_a}{\partial b} = \frac{1}{2a+b}$
$\frac{\partial P_b}{\partial a} = \frac{1}{a+2b} \quad \frac{\partial P_b}{\partial b} = \frac{2}{a+2b} - 1$

3 Problem 3: Search (20 pts)

Consider the search problem below with start state S and goal state G. The transition costs are next to the edges, and the heuristic values are next to the states.



If we use Uniform-Cost Search:

(a) What is the final path for this search?

If we use Depth First Search, and it terminates as soon as it reaches the goal state:

(b) What is the final path for this DFS search? If a node has multiple successors, then we always expand the successors in increasing alphabetical order.

If we use A* search:

(c) What is the final path for this A* search?

(d) Is the heuristic function in this example admissible?

4 Problem 4: Hill Climbing, Simulated Annealing and Genetic Algorithm (20 pts)

An N-Queens problem is to place N Queens on an NxN chess board such that no queen attacks any other (a queen can attack any other piece in the same row, column or diagonal). Let's consider one slightly efficient complete-state formulation as below:

- State: All N queens are on the board, one queen per row and per column. In this way, we only need to worry about the attacks along the diagonal, and this simplifies the evaluation function calculation.
- Evaluation: Number of *nonattacking* pairs of queens in this state.
- Successor Function: Swap of *adjacent* columns. For example, swap (1,2) means swap the column#1 and column#2

Let's study the 5-Queens problem:

(a) Given the definition above, how many states are there in total?

Number of states:______

```
. . Q . .        . . Q . .   . Q . . .   . . . Q .   . . Q . .
Q . . . .        . Q . . .   Q . . . .   Q . . . .   Q . . . .
. Q . . .        Q . . . .   . . Q . .   . Q . . .   . Q . . .
. . . Q .        . . . Q .   . . . Q .   . . Q . .   . . . . Q
. . . . Q        . . . . Q   . . . . Q   . . . . Q   . . . Q .
```

Figure 1: Initial state

Figure 2: Successors

(b) If we carry out steepest ascent hill-climbing starting from the initial state in Figure 1, what is the final state, and is it a solution? (The evaluation function values for the initial state and its four successors are given as below.)

InitState: Eval = 8
swap(1,2): Eval = 4
swap(2,3): Eval = 6
swap(3,4): Eval = 6
swap(4,5): Eval = 6

(c) Consider the relations between simulated annealing and variants of hill-climbing in a general setting:

When T=∞, simulated annealing is:_______

- A. steepest ascent
- B. stochastic hill climbing

- C. first-choice hill climbing
- D. random-restart hill climbing
- E. none of the above

When the temperature decay rate = 1, simulated annealing is ____

- A. steepest ascent
- B. stochastic hill climbing
- C. first-choice hill climbing
- D. random-restart hill climbing
- E. none of the above

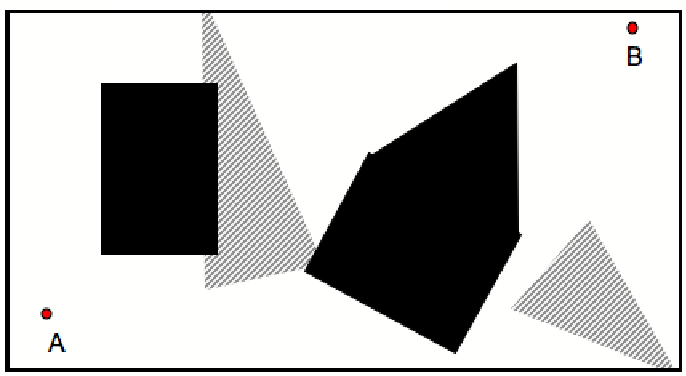
5 Problem 5: Search and Motion Planning (20 pts)

Consider the 2-D environment shown below. The environment contains obstacles (shown in black) and is observed by a set of surveillance cameras. The footprint of the fields of view of the camera is shown as shaded polygons. We are interested in how a path may be planned from point A to point B under different levels of robot shyness: don't care about the cameras; avoid the cameras at all cost; and try to stay out of the cameras' fields of view.

(a) Describe how the visibility graph algorithm can be used to find a path for the robot from point A to point B as quickly as possible, ignoring the cameras. Draw the output of your algorithm on the diagram below.

Description:

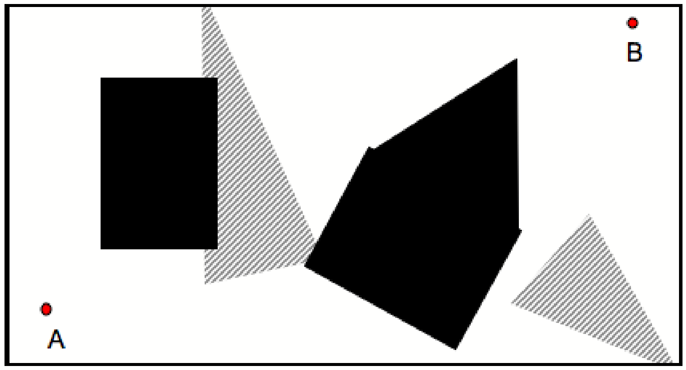
Output:



(b) Describe how the visibility graph algorithm can be used to find a path for the robot from A to B such that the robot is not seen by any of the cameras.

Description:

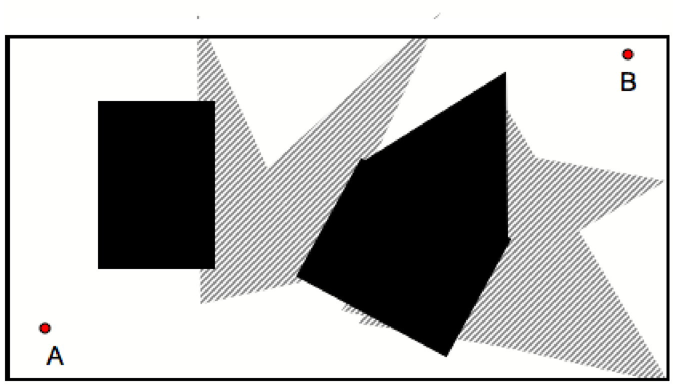
Output:



(c) In the environment shown below, it is now impossible to hide from the cameras all the time. Imagine an algorithm that will find a path for the robot from A to B such that the time during which the robot is seen by the cameras is minimized. Important: There is not a single possible answer to this, and it may not be possible to design an algorithm which is guaranteed to find the "optimal" path. We are just asking you to propose a sensible algorithm.

Description:

Output:



6 Problem 6: Constraint Satisfaction (20 pts)

Consider the perennial problem of scheduling classrooms in Wean Hall. We have 4 instructors ($I1, I2, I3, I4$) and 3 rooms ($R1, R2, R3$). We need to assign rooms to instructors. We assume that the instructors need the rooms at the following times:

I1: 9am to 11am
I2: 10am to 2pm
I3: 1pm to 5pm
I4: 1pm to 3pm

We assume that a room can be use by only one instructor at a time and that room $R3$ is too small for instructor $I1$ and that rooms $R2$ and $R3$ are too small for instructor $I3$.

(a) Show the search with forward checking by writing the domain for each variable at every step in the table below. Write the variable instantiated at each step of the search in the left column and the corresponding value domain for each the variables in the remaining entries of the table. Use the variable ordering (I1,I2,I3,I4) and the value ordering (R1,R2,R3).

Variable Instantiated	I1	I2	I3	I4
Initial Domains	R1,R2	R1,R2 R3	R1	R1,R2 R3

(b) Can the problem be solved by constraint satisfaction alone without backtracking?