15-381 Spring 2007
Midterm Exam

Spring 2007
March 8

Name:____ Andrew ID:________________________

- This is an open-book, open-notes examination. You have 80 minutes to complete this examination.
- Unless explicitly requested, we do not need to see the details of how you arrive at the answer. Just write the answer and do not waste time writing the details of the derivation.
- 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!

Question	Points
1	/20
2	/20
3	/20
4	/20
5	/20
5.3	/15 - Extra Credit
TOTAL	

Question 1 Local Search

You are given sentences in a language that you know nothing about, where the word order has been lost. Given the words, you are requested to put them back in a sequence, that has a sensible meaning.

For example, in English language, the words {**fun, exams, term, are, mid**} can be put together in this valid ordering: **"mid terms exams are fun"**.

As stated, the language used is completely unknown to you. Luckily, you can use an oracle that assigns a score to every sequence you introduce, according to how ridiculous it is (for example, in English, the sequence "exams mid term" is considered less ridiculous than "term exams mid").

1. (4 points) Given a set of n words, what is the problem search space size? (you may assume that all words are unique.)

   n!

2. (4 points) Define a neighborhood (a 'moveset', that can be reached in a single step of search). Give two examples of neighbors for the sequence "fun exams term are mid".

   There are many eligible neighborhoods. For example, the set of sequences where the positions of two words are swapped. According to this definition, neighboring sequences are: "exams fun term are mid", "term exams fun are mid" etc.

3. (4 points) Will *hill-climbing* search always find a valid sentence?

   No. Hill-climbing search is likely to converge to a local optimum.

4. (4 points) Will *stochastic hill-climbing* search always find a valid sentence?

   No. Stochastic hill-climbing search is less prone to converge to an arbitrary local optimum and is more likely to find a better solution. However, it is not guaranteed to find the global optimum. Theoretically, it should find the global optimum as running time reaches infinity (note, however, that this is also the case for brute force search).

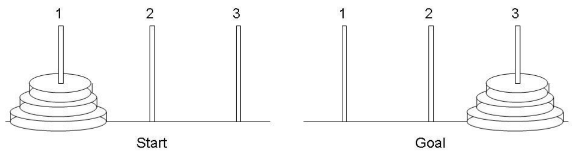
5. (4 points) In order to apply a genetic algorithm search, you'd need to define a cross-over operation. Can you suggest a cross-over operation here? If yes, describe your suggestion. If not, explain why in at most two sentences.

   In this problem (similarly to the TSP), the solution should include every word mention once. The operation of switching the left/right parts of two parents, cutting the sequences at some position t, may yield illegitimate children that violate this requirement.

   There are several ways of modifying the cross-over operation to generate valid children. For example: choose the first sub-sequence from parent a; then, concatenate the remaining words in the order that they appear in parent b to generate a valid child.

Question 2 Search

The Tower of Hanoi is a game in which a stack of disks of varying sizes are moved from one peg to another. The diagram below shows the start and goal states for the 3 disk/3 peg version.

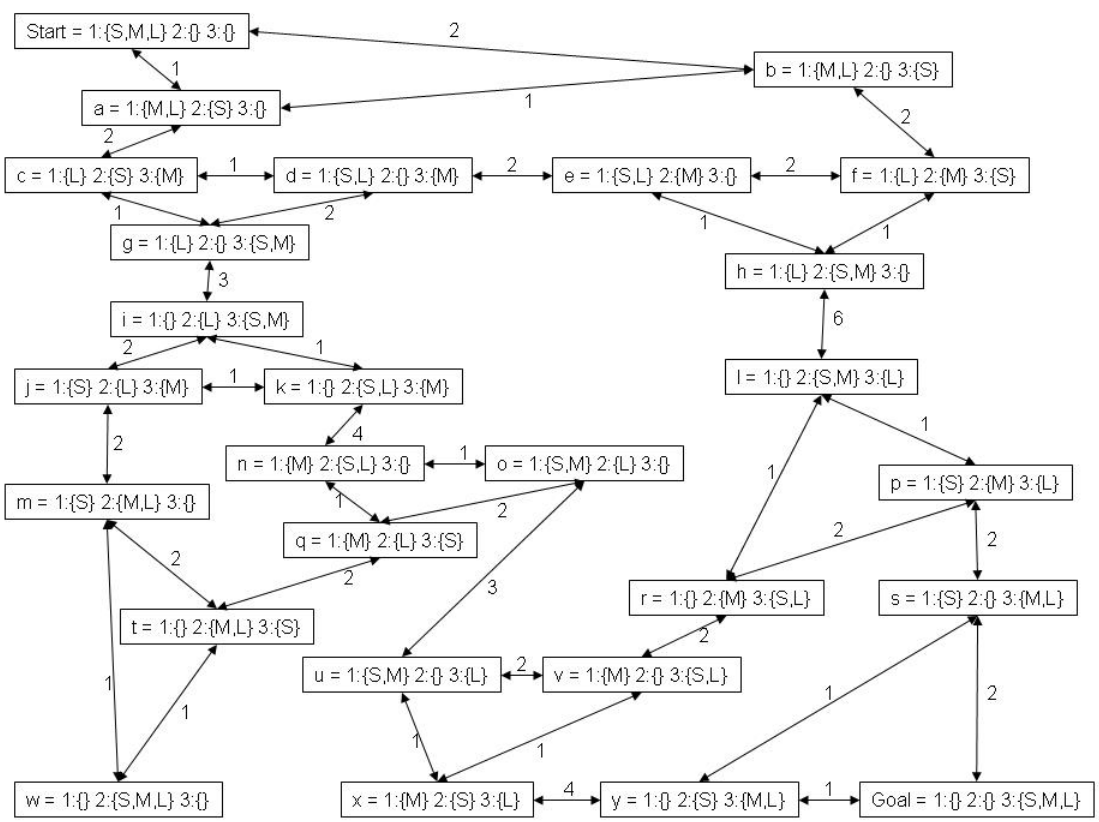


The rules of the game are that a disk may only be moved if it is on the top of a stack, and it can only be placed on an empty peg or a larger disk.

To formulate this as a search problem, let the disks be called S (small), M (medium), and L (large). We will represent the states 3 ordered sets (one for each peg, called 1, 2, and 3), where the first element of the set is the top disk, and the last element the bottom disk. (Obviously, the sets can also be empty.)

Let us further assume that S weighs 1 unit, M weighs 2 units, and L weighs 3 units. The cost of a move is the weight of the disk, multiplied by the distance of the move (1 for a neighboring peg, 2 for a peg 2 steps away).

The search tree, with costs, is shown below:



1. (6 points) Show the first 6 nodes expanded the uniform cost search algorithm on this problem (do not count the expansion of the *Start* node). Show the priority queue at each step. Assume that ties are broken by alphabetical order of the node names.

Node To Expand	Priority Queue After Expansion
Start	(a,1) (b,2)
a	(2,b) (5,c,a)
b	(4,f,b) (5,c,a)
f	(5,c,a) (5,h,f,b) (6,e,d,c,a)
c	(5,h,f,b) (6,d,c,a) (6,e,d,c,a) (6,g,c,a)
h	(6,d,c,a) (6,e,d,c,a) (6,g,c,a) (11,l,h,f,b)
e	(6,e,d,c,a) (6,g,c,a) (11,l,h,f,b)

There is a typo in the state space: the arc from a to c should have cost 4. Students were advised of this typo in class, and this is the correct solution taking that into account. However, full points were also given for the solution using the typo.

2. (8 points) Answer true/false for each of the following questions about this search problem.

   (a) Depth first search will find an optimal path with respect to the cost of the path.

      False

   (b) Depth first search will find an optimal path with respect to the number of steps in the path.

      False

   (c) Breadth first search will find an optimal path with respect to the cost of the path.

      Both answers were accepted. BFS is not guaranteed to find the optimal-cost solution because it does not take cost into account. However, in this particular problem, the BFS solution happens to be the optimal-cost path.

   (d) Breadth first search will find an optimal path with respect to the number of steps in the path.

      True

3. (6 points) Suppose we wish to use informed search, and we are considering what heuristic to use.

   (a) Which of these heuristics are admissible? Answer yes/no for each one.

      i. The number of disks on peg 3.

         No

      ii. The number of disks NOT on peg 3.

         Yes

      iii. The weight of the disks NOT on peg 3.

         Yes

      iv. The weight of the disks NOT on peg 3 multiplied by their distances from peg 3.

         Yes

   (b) Which of the admissible heuristics is best and what property makes it the best?

      Heuristic iv is the best because it dominates the other admissible heuristics (it is greater than or equal to them for all states). One point was deducted if you didn't say that it dominates or use the definition of a dominating heuristic in your answer.

Question 3 Constraint Propagation

Consider the following CSP: There are three variables A,B, and C which can take two values 1 and 2. There are the following constraints on the variable assignments, where each constraint indicates that if one variable is assigned its specified value the second variable CANNOT be assigned its specified value and vice-versa:

$$A = 1, B = 1$$
$$A = 1, C = 1$$
$$B = 2, C = 2$$
$$B = 1, C = 2$$

For both parts of this question use an alphabetic variable order and numeric value order (1 before 2). Assume that an expansion occurs whenever a variable is assigned a value from its domain.

1. (2 points) What is the first satisfying assignment found using depth-first-search with backtracking?

   $A = 2, B = 1, C = 1$

2. (4 points) What is the order of assignments made using depth-first-search with backtracking for a single satisfying assignment?

   $A = 1, B = 1, B = 2, C = 1, C = 2, A = 2, B = 1, C = 1$

3. (4 points) What is the order of assignments made using depth-first-search with backtracking AND forward checking for a single satisfying assignment?

   $A = 1, B = 2, A = 2, B = 1, C = 1$

4. For all of the following statements you must indicate whether or not the statement is true or false.

   (a) (2 points) Using constraint propagation as part of DFS with backtracking never increases the number of node expansions in the search.

   True. Constraint propagation will only reduce the domains of nodes, so it can't cause an increase in node expansions.

   (b) (2 points) Using constraint propagation as part of DFS with backtracking never increases the time complexity of the search.

   False. Constraint propagation can be of arbitrary time complexity, and in some cases can be more expensive than conducting the search.

   (c) (3 points) Value and variable ordering can decrease the number of expansions performed when using search to determine that a CSP has no solution.

   False. To determine that there is no solution we must search the entire space created by DFS with backtracking - the order that we assign variables and values doesn't change the size of this space.

   (d) (3 points) Given that a number of search assignments have been made as part of DFS with backtracking and constraint propagation, if constraint propagation leaves a node with an empty domain there is no solution to the CSP.

   False. This branch of the search is invalid and we need to backtrack, but as we have made some number of arbitrary assignments during search a solution to the CSP may still exist.

Question 4 Robot Motion Planning

1. (5 points) The initial visibility graph is shown with dotted lines. The path found from start (S) to goal is shown with solid lines.

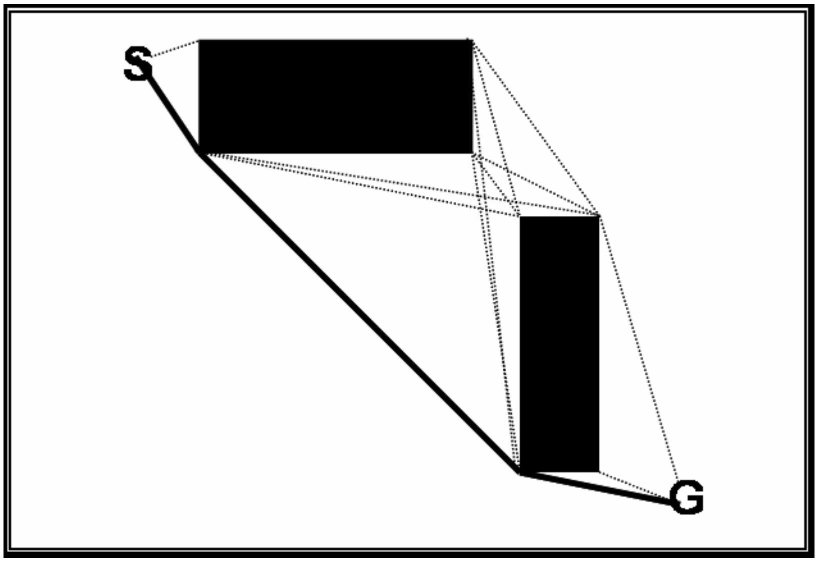


Figure 1: Visibility graph solution

2. The Rapidly Exploring Random Tree is shown with solid lines.

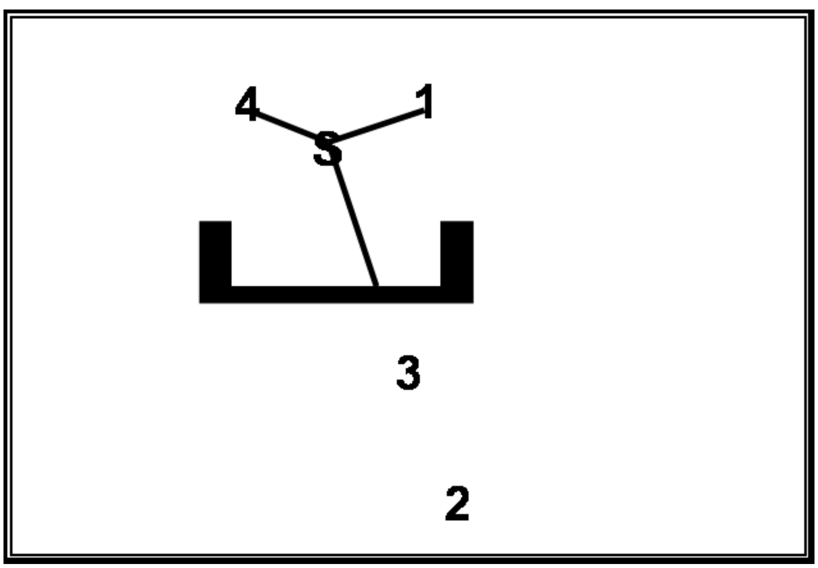


Figure 2: RRT solution

3. (5 points) The probabilistic roadmap contains 0 edges because all $K = 1$ nearest neighbor edges cross an obstacle. There is no path from start to goal.

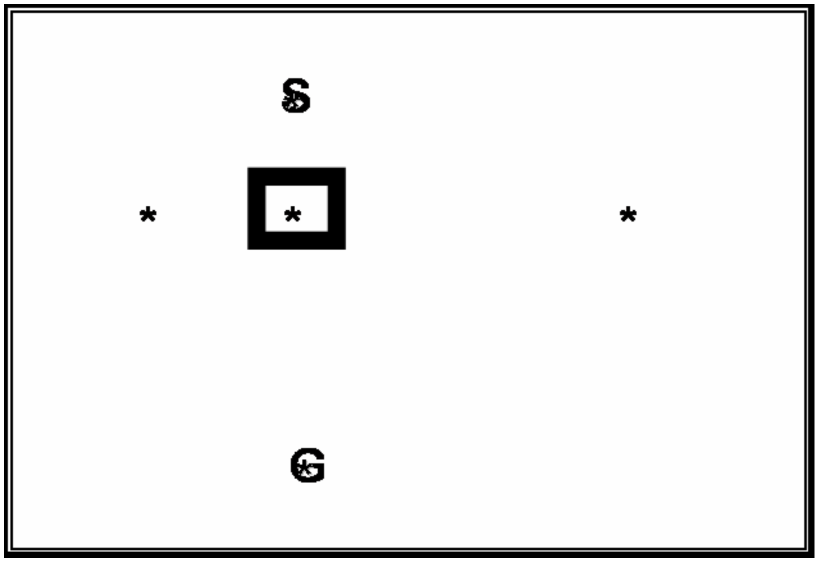


Figure 3: Probabilistic Roadmap k=1 solution

4. (5 points) The probabilistic roadmap when $K = 2$ is shown with dotted lines. The path found from start (S) to goal is shown with solid lines.

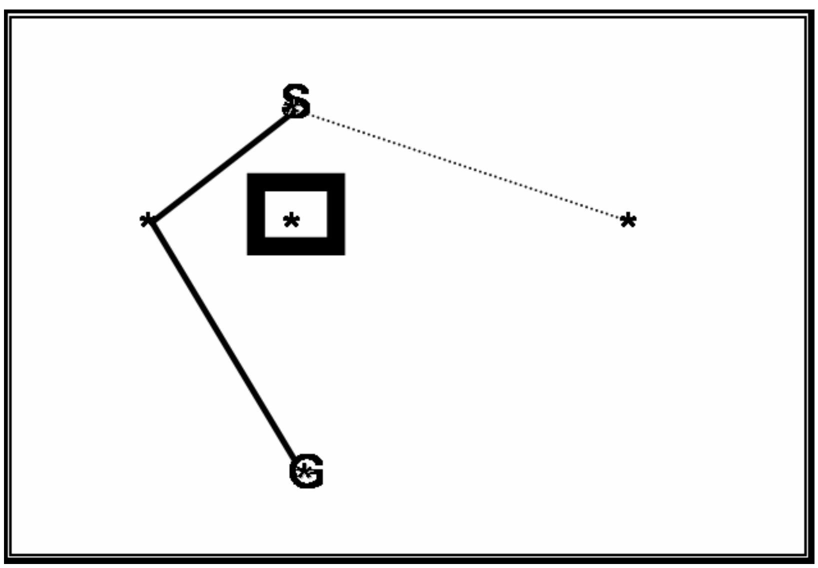


Figure 4: Probabilistic Roadmap k=2 solution

Question 5 Game Theory

The purpose of this exercise is to show how the exact same payoff structure in a game can lead to different solutions depending on the observability of the moves, i.e., perfect information, hidden information, or chance elements.

	X	Y
X	3,2	1,1
Y	4,3	2,4

Consider the two-player game in which each player has the choice between 2 moves, X and Y. The players choose and execute their moves simultaneously, so that player A has to choose his/her move without knowing player B's move and vice-versa. This is a typical example of a hidden information game. The matrix form of the game is shown above.

1. (10 points) This game has one pure equilibrium. What are the players' moves in the pure equilibrium of this game?

   Both players choose Y.

2. (10 points) Suppose now that the players execute their moves sequentially. More precisely, player A executes his/her move first, and then player B chooses his/her move *after seeing* A's move. Write the matrix normal form for this new game. Show that this new game has two equilibria when written in matrix form. Given that player A moves first, what will be the moves selected by the two players?

   The matrix form is included below. The notation for player B is as follows: XX means that B plays X if A has played X and B plays X if A has played Y.

	XX	XY	YX	YY
X	3,2	3,2	1,1	1,1
Y	4,3	2,4	4,3	2,4

   The two equilibria are (X, XY) and (Y, YY).

   Since A plays first, he will choose the solution with the highest payoff for A. Therefore, A plays X and B plays X.

3. (15 points, ***Extra Credit***) Suppose now that player B is a little confused and he/she does not accurately observe A's move. More precisely, if A plays X, then B observes X with probability $1-\epsilon$ and Y with probability ϵ. Conversely, if A plays Y, then B observes Y with probability $1-\epsilon$ and X with probability ϵ ($\epsilon > 0$.) Write down this new version of the game in matrix form. There is one *pure* equilibrium for this new game. What are the players' strategies at the equilibrium?

   Hint: Pretend that there is a chance move after player A moves, like a roll of a die before B moves.

   Note: We are *not* asking to find the mixed equilibria. Only the one pure equilibrium in this example.

The matrix form is:

	XX	XY	YX	YY
X	3,2	$3-2\epsilon$,$2-\epsilon$	$1+2\epsilon$,$1+\epsilon$	1,1
Y	4,3	$2+2\epsilon$,$4-\epsilon$	$4-2\epsilon$,$3+\epsilon$	2,4

The only equilibrium is now (Y, YY) and both players choose Y.

Note: It is interesting that inserting an element of chance, no matter how small, changes the strategy chosen by the players. In fact, it reverts back to the moves selected for the original game (Y for both players!).