Exam 1– Spring 2019

Name: Hoang Do

This is a closed book exam. Notes are not allowed. Computational aids are not allowed. You have 75 minutes.

If a question is not clear to you, state your assumption and move on!

Problem	Topic	Possible Points	Score Earned
1	Search Formulation	9	9
2	A* and DFS	11	11
3	Questions merengue	15	15
4	Hill Climbing	15	2
5	Genetic Algorithms	15	14
6	Adversarial Search	10	2
7	Constraint Satisfaction Problems	15	8
	TOTAL	90	61

1. Search Formulation (9 pts).

Pacman and Ms. Pacman are lost in an *NxN* maze and would like to meet; *they don't care where.* In each time step, *both* simultaneously move in one of the following directions: {NORTH, SOUTH, EAST, WEST, PAUSE}. They do *not* alternate turns. You must devise a search which positions them together, somewhere, in as few time steps as possible. Passing each other does not count as meeting; they must occupy the same square at the same time.

(a) Formally state this problem as a single-agent state-space search problem (4 pts).

States: Px, Py are directions of Pacman.
Mx, My are directions of Ms Pacman 1

Maximum size of state space:

N^4 1

Maximum branching factor:

$5^2 = 25$ 1

Goal test:

$(Px = Mx) \wedge (Py = My)$ 1

(b) Give a non-trivial admissible heuristic for this problem (2 pts).

$h_x = \dfrac{\text{Manhattan-direction}(P, M)}{2}$ 1

(c) Circle all of the following graph search methods that are guaranteed to output optimal solutions to *this problem* (3 pts):

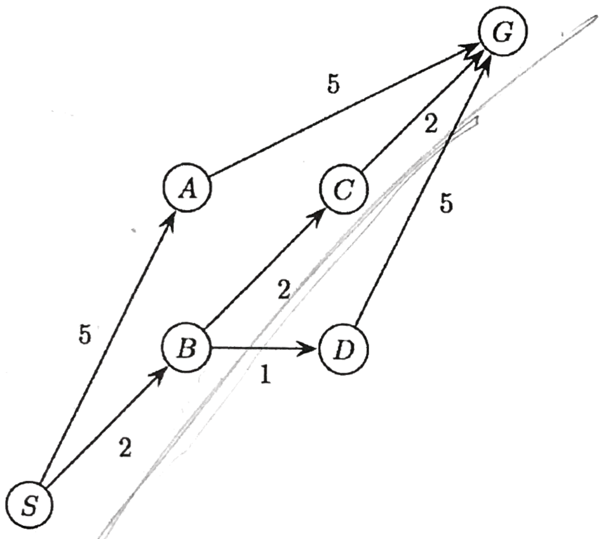
(i) DFS

(ii) BFS

(iii) A* (with a heuristic that returns zero for each state) 3

2. **A* and DFS (11 pts)**

Consider the following search problem. The start state is S and the goal is G:



Node	h_0	h_1	h_2
S	0	5	6
A	0	3	5
B	0	4	2
C	0	2	5
D	0	5	3
G	0	0	0

Figure 1: Search Problem

(a) Which of the heuristics shown are admissible? Circle all that apply (3 pts):

h_0 h_1 h_2

3

(b) Give the path A* will return using each of the three heuristics. (Hint: you should not actually have to execute A* to know the answer in some cases.) (6 pts)

h_0: S -> B -> C -> G

h_1: S -> B -> C -> G 0

h_2: S -> B -> D -> G

(c) Briefly, explain the modification to graph search that allows DFS to perform with space complexity *O(bd)*, where *b* is the branching factor and *d* is the search depth (2 pts).

If DFS doesn't add all the states to the closet set, only search for the present node and its sibling on that path.

3. Questions merengue (10 points)

True/False: Each problem is worth 2 points. Incorrect answers are worth 0 points. Skipped questions are worth 0 point.

Circle *True* or *False*, or leave unmarked.

(a) *True* *False:* If one search heuristic $h_1(n)$ is admissible, and another one $h_2(n)$ is ***in***admissible, then $h_3(n) = \min(h_1(n), h_2(n))$ will be admissible.

(b) *True* *False:* If two admissible search heuristics $h_1(n)$ and $h_2(n)$ have the same average value, the heuristic $h_3(n) = \max(h_1(n), h_2(n))$ could give better A* efficiency than h_1 or h_2.

(c) *True* *False:* Depth-first graph search is always complete on problems with finite search graphs.

(d) *True* *False:* In A*, the first path to the goal which is added to the fringe will always be optimal.

(e) *True* *False:* An optimal solution path for a search problem with positive costs will never have repeated states.

Environments (5 points)

For each pair of environment characteristics, circle the ones that are necessary for an agent to effectively use search to solve a problem.

Fully observable *or* Partially observable

Deterministic *or* Stochastic

Episodic *or* Sequential

Static *or* Dynamic

Discrete *or* Continuous

4. Hill Climbing (15 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 *non-attacking* pairs of queens in this state.
- Successor Function: Swap of *adjacent* columns. For example, swap (1,2) means swap column#1 and column#2

Let's study the 5-Queens problem:

(i) Given the definition above, how many states are there in total? (3 points)

		Q		
Q				
	Q			
			Q	
				Q

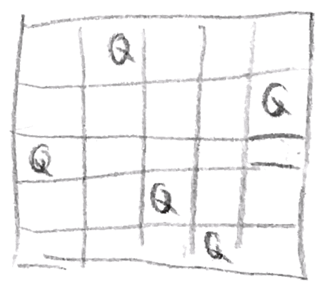
5^{20}

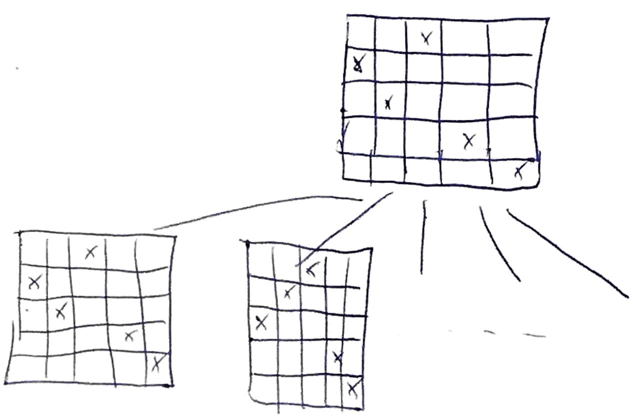
5!

Figure 2: 5-Queens Initial State

If we carry out steepest ascent hill-climbing starting from the initial state in Figure 3, 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.) (4 pts for answer, 8 pts for drawing the successors)

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





it is not a solution. since the last 3 row, queen's are still attack each others.

5. Genetic Algorithms (15 pts)

Suppose a genetic algorithm uses chromosomes of the form x = abcdefgh with a fixed length of eight genes. Each gene can be any digit between 0 and 9. Let the fitness of individual x be calculated as:

$$f(x) = (a + b) - (c + d) + (e + f) - (g + h)$$

and let the initial population consist of four individuals with the following chromosomes:

x1 = 6 5 4 1 3 5 3 2
x2 = 8 7 1 2 6 6 0 1
x3 = 2 3 9 2 1 2 8 5
x4 = 4 1 8 5 2 0 9 4

a) Evaluate the fitness of each individual, showing all your workings, and arrange them in order with the fittest first and the least fit last. (4 pts)

$f(x_1) = (6+5) - (4+1) + (3+5) - (3+2) = 9$
$f(x_2) = (8+7) - (1+2) + (6+6) - (0+1) = 23$
$f(x_3) = (2+3) - (9+2) + (1+2) - (8+5) = -16$
$f(x_4) = (4+1) - (8+5) + (2+0) - (9+4) = -19$

$\Rightarrow f(x_2) > f(x_1) > f(x_3) > f(x_4)$

4

b) Perform the following genetic operations:

a. Cross the fittest two individuals using a one-point crossover at the middle point (2pts)

x1 6 5 4 1 | 3 5 3 2
x2 8 7 1 2 | 6 6 0 1

$\Rightarrow$
c1 6 5 4 1 6 6 0 1
c2 8 7 1 2 3 5 3 2

2

b. Cross the second and third fittest individuals using a two-point crossover (points b and f) (2pts)

x1 6 5 | 4 1 3 5 | 3 2
x3 2 3 | 9 2 1 2 | 8 5

c1 2 3 4 1 3 5 8 5
c2 6 5 9 2 1 2 3 2

2

c) By looking at the fitness function and considering that genes can only be digits between 0-9 find the chromosome representing the optimal solution (i.e. with the maximum fitness). Find the value of the maximum fitness (4 pts).

$f_{max} \Leftrightarrow (a+b)+(e+f)$ max

$(c+d)+(g+h)$ min

$\Rightarrow (a+b+e+f) = 9+9+9+9$

$(c+d)+(g+h) = 0+0+0+0$

$\Rightarrow x_{max} = 99009900$

3

d) By looking at the initial population of the algorithm can you say whether it will be able to reach the optimal solution without any mutation? Explain your answer. (3 pts)

No, because in order to get x max, the first two number a, b of ~~x1~~, x2, x3, x4 must have 9; c, d, h must have 0. But in the Initial population x1, x2, x3, x4 don't have

3

6. Adversarial search (10 pts)

Figure 3 shows 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.

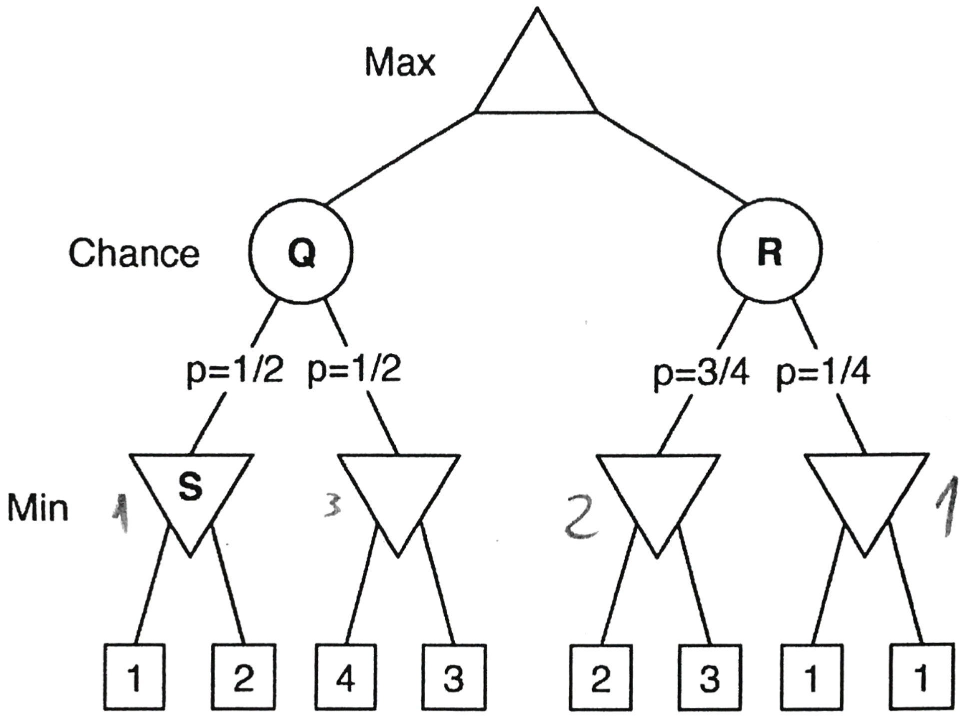


Figure 3: Expect minimax

a) What is the value of the node labeled S? (2 pts)

1

2

b) What is the expected value of node labeled Q? (show your computation) (2 pts)

1/2 chance for 1, and 1/2 chance for 3 $\left(\frac{1}{2} \times 1\right) + \left(3 \times \frac{1}{2}\right) = 2$

c) What is the expected value of node labeled R? (show your computation) (2 pts)

1/4 chance for 1, and 3/4 chance for 2

d) What is the expected value of the game? (show your computation) (2 pts)

If $Q = R = 1 \Rightarrow Max = 1$
If $Q = 3$, $R = 1$ or $R = 2 \Rightarrow Max = 3$
If $R = 2$, $Q = 1 \Rightarrow Max = 2$

e) Have you been provided with enough information such that you could perform alpha-beta pruning for this game? (2 pts)

It is not enough information to perform alpha-beta pruning for this game.

7. Constraint Satisfaction Problem (15 pts)

Here is an instance of a well-known logic puzzle called The Zebra Puzzle: There are five houses, each with a different color and inhabited by five persons of different nationalities, each of whom prefer a different brand of cigarette, a different drink, and a different pet. Moreover:

- ~~The Englishman lives in the red house.~~ E -> Red.
- The Spaniard owns the dog. S line dog.
- ~~Coffee is drunk in the green house.~~ C - Green.
- The Ukrainian drinks tea. U drink tea
- The green house is immediately to the right of the ivory house. Green — Ivory L
- The Old Gold smoker owns snails.
- ~~Kools are smoked in the yellow house.~~
- Milk is drunk in the middle house.
- ~~The Norwegian lives in the first house.~~
- The man who smokes Chesterfields lives in the house next to the man with the fox.
- Kools are smoked in the house next to the house where the horse is kept.
- The Lucky Strike smoker drinks orange juice.
- The Japanese smokes Parliaments.
- ~~The Norwegian lives next to the blue house.~~

Query: Using the constraints given above, who owns the zebra and who drinks water? **Note that this assignment does not require you to answer the query.**

The goal is to model the The Zebra Puzzle as a CSP, using a constraint graph. In order to accomplish this, you must define a set of variables with their domain and then draw the constraint graph. For the purposes of this question, formulate the variables and their domain (10 pts) and then draw the constraint graph (5 pts) domain

Variables

	1	2	3	4	5
house-color	R	Y	G	I	B
drink	W	O	T	M	cof
Smoke	P	L	K	C	OG
Nation	E	S	U	J	N
Animal	D	F	Z	S	kept ?

= 8

Variable	Domain
R	{W,O,T,M}, {P,L,C,OG} {E} {F,Z,S,kept}
Y	{W,O,T,M}, {K}, {S,U,J,N}, {O,F,Z,S,kept}
G	{cof}, {P,L,C,OG}, {S,J,N}, {F,Z,S,kept}
I	{W,O,T,M}, {P,L,C,OG}, {S,U,J,N}, {F,Z,S,kept}
B	{W,O,T,M}, {P,L,C,OG}, {S,U,J,N}, {D,F,Z,S,kept}