3A07A74C-E898-4079-B6E7-0D95E90D9B36

cs561-s15exam1

#9 2 of 18

Q2 Score: **13.0**

1. General AI knowledge [15pts]

For each of the statements below, fill in the bubble **T** if the statement is always and unconditionally true, or fill in the bubble **F** if it is always false, sometimes false, or just does not make sense:

1	T	
2		F
3	T	
4		F
5	T	
6	~~T~~	
7	T	
8		F
9		F
10		F
11	T	
12	T	
13		~~F~~
14	T	
15	T	

1. A reflex agent that senses only partial information about the state cannot be rational.
2. There exists a deterministic task environment in which any agent is rational.
3. No agent is rational in an unobservable environment.

4. An agent function that specifies the agent print true when the percept is a Turing machine program that halts, and false otherwise, cannot be implemented.
5. Local beam search is an entirely deterministic algorithm.

6. Alpha-beta pruning may not find the optimal solution, because it prunes some branches.
7. Local beam search with one initial state and no limit on the number of states retained is Depth First Search.
8. Simulated annealing with $T = \infty$ at all times is a random-walk search

9. Mutation and crossover are methods to handle local extrema in Genetic Algorithms.
10. Even if we use a beam width of 1 and the same tie-breaking procedure, beam search and hill climbing are not complete.
11. If we have an admissible heuristic function, then when a node is expanded and placed in CLOSED, we have always reached it via the shortest possible path. $f(n) = h(n)$
12. If greedy search finds some goal node in finite time, then depth-first search will also find a goal node in finite time.
13. If depth-first search finds some goal node in finite time, then beam search will also find a goal node in finite time.
14. The primary purpose of the CLOSED list in search is to produce the path from the goal node found back to the initial state.
15. Genetic algorithm with population size $N = 1$ is the same as local beam search with k=1.

mutation → child → valid > 1

↳ DFS

Q3 Not graded

34D3B0F7-565D-4455-BFA6-C0D211C18F29

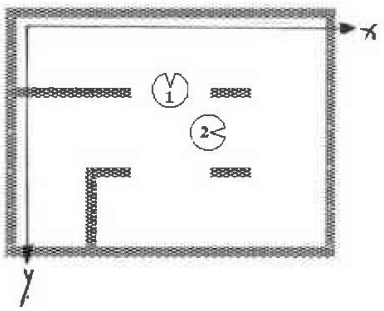
cs561-s15exam1

#9 4 of 18

Q4 Score: **3.0**

2. Search Algorithms Concepts [10pts]

Pacman (1) and Ms. Pacman (2) are lost (each at random initial state) 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, STOP}. They do not alternate turns. You must devise a plan that 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) [4pts] Formally state this problem as a **single-agent** state-space search problem.

States: Assume the original point is the left-top square in the maze, use coordinates to represent the location of ① and ②. Pacman in the location (x_1, y_1), Ms. Pacman in the location (x_2, y_2). $x_1, x_2, y_1, y_2 \in [0, N-1]$ As shown in the graph. ✓

Maximum size of state space: N^4 (since each has N^2 different states if there is no wall) ✓

Maximum branching factor: 5^4 ✗

Goal test: $x_1 == x_2$ && $y_1 == y_2$ ✓

Q5 Not graded

84A3B33F-A49B-4568-80C1-1CDEF71ACD62

cs561-s15exam1

#9 6 of 18

Q6 Score: **4.0**

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

If for node n, the positions are $(x_1(n), y_1(n))$ and $(x_2(n), y_2(n))$.
Then we can choose $h(n) = \left(|x_1(n) - x_2(n)| + |y_1(n) - y_2(n)|\right)/2$.
(e.g. assume positions are (1, 6) and (1, 10), then $h = 2$) ✓

(c) [2 pts] Circle all of the following graph search methods which are guaranteed to output optimal solutions to this problem:

(i) DFS

(ii) BFS

(ii) UCS

(iv) A* (with a consistent and admissible heuristic)

(v) A* (with heuristic that returns zero for each state) = UCS

(vi) Greedy search (with a consistent and admissible heuristic)

(d) [2 pts] If h1 and h2 are admissible, which of the following are also guaranteed to be admissible? Circle all that apply:

(i) h1 + h2

(ii) h1 * h2

(iii) max(h1 , h2) ✓

(iv) min(h1 , h2)

(v) $(\alpha)h1 + (1-\alpha)h2$, for $\alpha \in [0,1]$

Q7 Not graded

35CBD75F-1544-4907-8BC6-F932487C5A1F

cs561-s15exam1

#9 8 of 18

Q8 Score: **24.0**

3. Comparing Search Strategies [30pts]

Consider the search space on the following page, where S is the start node and G1 and G2 satisfy the goal test. Arcs are labeled with the cost of traversing them and the estimated cost to a goal is reported inside nodes.

For each of the following search strategies, indicate which goal state is reached (if any) and list, in order, all the states of the nodes **popped off** of the OPEN queue. When all else is equal, nodes should be removed from OPEN in alphabetical order.

You should not expand nodes with states that have already been visited. Note how the arcs in the figure are oriented, which means that you can only go from one state to another if the arrow points from the first to the second. For example, you can go from S to C (i.e., C is a successor of S) but not from C to S (i.e., S is not a successor of C).

(a) [6 pts] Greedy search

States popped off OPEN: S B F G_1 ✓ Goal state reached: G_1

(b) [8 pts] Uniform Cost search

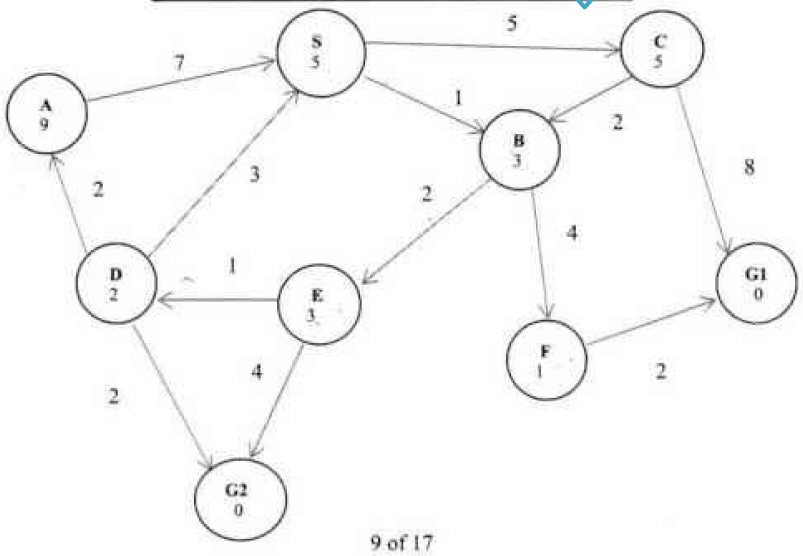
States popped off OPEN: S B E D C F A G_2 ✓ Goal state reached: G_2

(c) [8 pts] Iterative Deepening Search

States popped off OPEN: S B C ✗ Goal state reached: G_1

(d) [8 pts] A* Search

States popped off OPEN: S B E D F G_2 ✓ Goal state reached: G_2



Q9 Not graded

53FDCC8A-7F50-4D22-9F16-417FD0891C7D

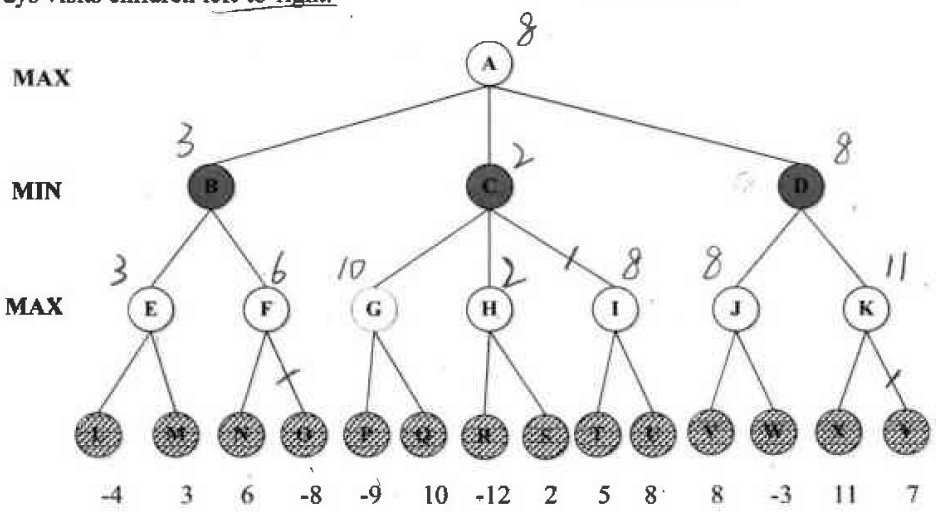
cs561-s15exam1

#9 10 of 18

Q10 Score: **12.0**

4. Game Playing [15pts]

Consider the following game tree in which the evaluation function values are shown below each leaf node. Assume that the root node corresponds to the maximizing player. Assume that the search always visits children left-to-right.



(a) [4 pts] Compute the backed-up values computed by the minimax algorithm. Show your answer by writing values next to the appropriate nodes in the above tree.

(b) [6 pts] Which nodes will not be examined by the alpha-beta pruning algorithm? List the letter of the prune nodes.

O, T, U, Y

missing I

(c) [5 pts] Can the nodes B, C, D be expanded in a different order to improve the number of nodes pruned? If yes, list the new order of nodes B, C, D and the letter of the prune nodes, or if no, explain why not.

Yes. DBC or DCB

Pruning nodes: Y, N, O, T, U

missing I,T You should write all the nodes in the subtree not just the leaves.

Q11 Not graded

3BDCF489-97BC-4EA4-AA47-06CD60B7A873

cs561-s15exam1

#9 12 of 18

Q12 Score: **15.0**

5. Constraint Satisfaction [15pts]

You are designing a menu for a special event. There are several choices, each represented as a variable: (A)ppetizer, (B)everage, main (C)ourse, and (D)essert. The domains of the variables are as follows:

A: (v)eggies, (c)aviar

B: (w)ater, (s)oda, (m)ilk

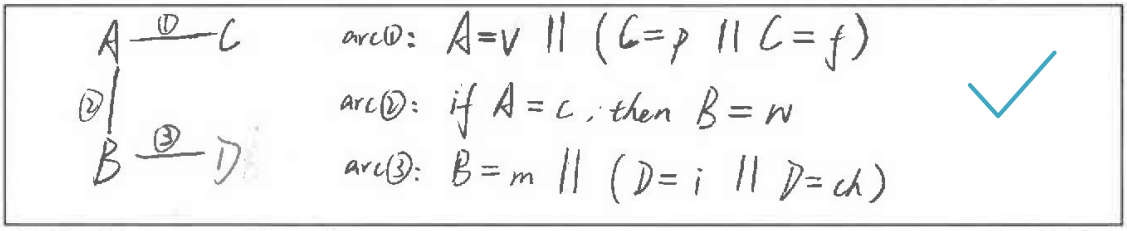
C: (f)ish, (b)eef, (p)asta

D: (a)pple, (i)ce cream, (ch)eese

Because all of your guests get the same menu, it must obey the following dietary constraints:

(i) Vegetarian options: The appetizer must be veggies or the main course must be pasta or fish (or both).
(ii) Total budget: If you serve the caviar, you cannot afford any beverage other than water.
(iii) Calcium requirement: You must serve at least one of milk, ice cream, or cheese

(a) [5 pts] Draw the constraint graph over the variables A, B, C, and D.



(b) [4 pts] Imagine we first assign A=c. What values remain for each variable after forward checking?

B: w
C: p, f
D: a, i, ch

(c) [4 pts] Again imagine we first assign A=c. What values remain for each variable after arc consistency has been enforced.

B: w
C: p, f
D: i, ch

(d) [2pts] Give a solution for this CSP or state that none exists.

A=c B=w C=p D=ch

Q13 Not graded

546159FF-01C4-49D5-A069-7215221DA97A

cs561-s15exam1

#9 14 of 18

6: 4

7: 2

4: 2

5: 1

inside 6: $6 \times 6 \times 6 = 216$

6+7: $6 \times 7 \times 4 \times 2 = 336$

6+4: $6 \times 4 \times 4 \times 2 = 192$

6+5: $6 \times 5 \times 4 = 120$

inside 7: $7 \times 7 = 49$

7+4: $7 \times 4 \times 2 \times 2 = 112$

7+5: $7 \times 5 \times 2 = 70$

inside 4: $4 \times 4 = 16$

4+5: $4 \times 5 \times 2 = 40$

42 × 8 = 336

32 × 6 = 192

16 × 7 = 112

216
336
192
120
49
112
70
16
+ 40
1151

Q14 Score: **5.0**

6. Local Search [15pts]

Sudoku problem, requires that the same single integer may not appear twice in the same row, column or in any of the nine 3×3 subregions of the 9x9 playing board. To simplify our problem, we do not consider the subregions which means you can have the same integers in any of the nine 3×3 subregions. Each state is represented by a 9*9 2D-array of numbers. The only action we can do is to swap two numbers in the same column.

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

1: 6
2: 7
3: 6
4: 7
5: 6
6: 4
7: 6
8: 4
9: 5

6: 4
7: 2
4: 2
5: 1

(a) [2pts] How many neighbor states for this sample state?

$\frac{9\times 9\times 8}{2} = 324$ ✓

(b) [2pts] What is the total size of the search space, i.e. how many possible states are there in total?

$(9!)^9$ ✓

(c) [2pts] Suppose all black numbers are fixed. Only gray numbers can be swapped. How many neighbor states for this sample state? how many possible states are there in total?

neighbor: $\frac{4\times 3+6\times 5+8\times 7+6\times 5+3\times 2+6\times 5+8\times 7+6\times 5+4\times 3}{2} = 131$ ✓

possible: $(4!)^2\cdot(6!)^4\cdot(7!)^2\cdot 5!$ ✗

Q15 Not graded

78945F44-FCA7-4C8A-81FF-861E0EE48FE8

cs561-s15exam1

#9 16 of 18

Q16 Score: **9.0**

Look at the state below. It is a simplified Sudoku problem that requires the same single integer may not appear twice in the same row or column. You decide to use a hill-climbing algorithm to solve this problem.

	1	1	7	1	8	6	5	1	1
	2	3	4	2	1	7	3	6	2
	6	4	2	3	3	5	6	9	3
	3	2	3	4	9	4	7	2	7
	4	6	9	9	5	1	4	3	4
	8	8	5	6	7	2	8	5	5
	5	5	6	7	6	8	9	4	9
	7	7	8	8	2	3	1	7	6
	9	9	1	5	4	9	2	8	8

(d) [4pts] Design a heuristic function for your hill-climbing algorithm. (Your h function for goal state should be 0.)

This heuristic function returns half of the sum of the number of same integers that appear in the same column or row. For instance, the last row should be $\frac{3+2}{2} = 2.5$. For this state, $h(n) = 23$

(e) [5pts] Base on your designed heuristic function, write an example of a valid action you may choose from this state.

1	1	7	1	8	6	5	1	1
2	3	4	2	1	7	3	6	2
6	4	2	3	3	5	6	9	3
5	2	3	4	9	4	7	2	7
4	6	9	9	5	1	4	3	4
8	8	5	6	7	2	8	5	5
3	5	6	7	6	8	9	4	9
7	7	8	8	2	3	1	7	6
9	9	1	5	4	9	2	8	8

The number circled are exchanged based on initial state. Now $h(n) = 23$

Q17 Not graded

144740C6-AC1D-4474-8A3E-258258AEDC37

cs561-s15exam1

#9 18 of 18