CPE/CSC 480 ARTIFICIAL INTELLIGENCE

MIDTERM PART 2

SECTION 1

FALL 2003

PROF. FRANZ J. KURFESS

CAL POLY, COMPUTER SCIENCE DEPARTMENT

This is the second Fall 2003 midterm exam for the CPE/CSC 480 class. You may use textbooks, course notes, or other material, but you must formulate the text for your answers yourself. The use of calculators or computers is allowed for viewing documents or for numerical calculations, but not for the execution of algorithms or programs to compute solutions for exam questions. The exam time is 60 minutes.

Student Name:

Signature: **Date:**

PART 1: MULTIPLE CHOICE QUESTIONS

Mark the answer you think is correct. Unless otherwise noted, there is only one correct answer. Each question is worth 3 points.

a) The *IDA* algorithm* is a variation of A* with the following properties?
- ❑ the nodes within a given contour are explored in a depth-first manner, and the contour is expanded step by step
- ❑ in addition to the current path, it stores the information about the best alternative to explore in case the current path doesn't lead to the goal
- ❑ it drops the least promising nodes from the fringe (search queue) when it runs out of memory
- ❑ it utilizes contours to reduce the number of nodes to explore

b) What is the most important difference between *local* and conventional (uninformed and informed) search methods?
- ❑ the search algorithm investigates only nodes that are reachable from the current node
- ❑ a solution has to be found, but the actual path to the solution is irrelevant
- ❑ it works better in continuous environments
- ❑ the analogy to landscapes with hills and valleys is more appealing than the one with trees

c) What is the best characterization of *mutation* in genetic algorithms?
- ❑ an individual's state description is modified randomly
- ❑ the state description of the parents is split at the crossover point, and they exchange parts
- ❑ useful components within individuals are preserved across generations
- ❑ the duplication of a randomly selected individual

d) What constitutes a *state* in a constraint satisfaction problem?
- ❑ the set of all variables used in the specification of the problem
- ❑ the set of all variables together with their respective potential values
- ❑ an assignment of values to some or all variables
- ❑ an assignment of values to *all* variables

e) What is the *contingency problem* in the context of game-playing programs?
- ❑ a degree of uncertainty, introduced by the presence of an opponent or by chance elements
- ❑ the outcome of a move may not be visible due to search limitations
- ❑ the need for arbitration (e.g. by a referee) in some types of games
- ❑ the elimination of branches that will never be explored

PART 2: SHORT QUESTIONS

In this part of the exam, you should answer the questions in one paragraph per aspect.

a) What is the importance of using contours for the A* search method?

[5 points]

b) What happens in the minimax algorithm if MIN doesn't't' select the best move (from MIN's perspective), but a subotpimal one?

5 points

PART 3: MAZE SEARCH

In this scenario, an agent is trying to traverse a maze from the starting point **S** to the goal point **G**. At each step, the agent can move in one of the four compass directions; each move, independent of the direction, costs the agent one cost unit. The agent always considers alternative moves clockwise, i.e. in the following order:

1. Move North
2. Move East
3. Move South
4. Move West

In the following parts, you need to apply different search algorithms to solve this navigation problem. Number the squares in the order the agents visits the squares, starting with 0 at the starting point.

In those algorithms that use it, calculate the path cost on the basis of one cost unit per move.

The heuristics to use in the respective algorithms is the difference between the horizontal position of the current node and the goal node, plus the difference in the vertical position of the current node $[x_n,y_n]$ and the goal node $[x_g,y_g]$, adjusted by a small value to break symmetries:

$$h([x,y]) = (|x_g - x_n|) + 0.99(|y_g - y_n|)$$

So, for the starting point, node [1,8], the heuristics with respect to the goal, node [5,3], is (5-1) + 0.99*(8-3) = 4 + 4.95 = 8.95.

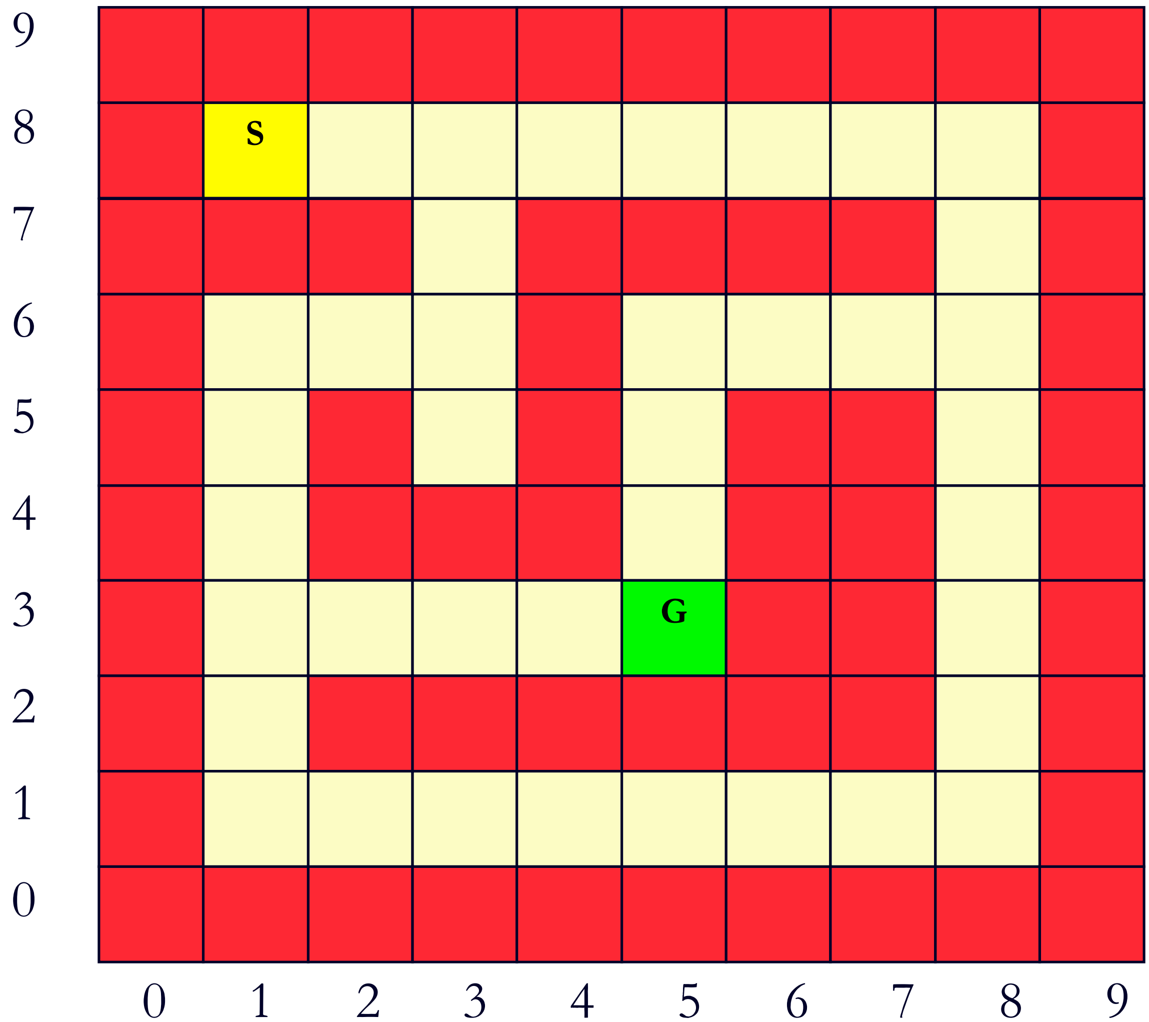
For the following algorithms, you need to do the following tasks

- Mark the sequence in which the nodes are visited in the maze.
- Draw the corresponding search tree.
- Fill in the table with the information about the search trace.

You can perform the algorithms in an off-line manner. This means that you can "jump" from the current node to the next node in the queue, without backtracking through already visited nodes. You can also assume that nodes already in the queue or previously visited will not be examined again, thus avoiding cycles.

a) Traverse the maze from the starting point **S** to the goal **G** according to the *Greedy Best-First Search* method. Calculate the value of the heuristic for each accessible tile in the maze and write it in the tile, mark the path selected by the agent, and record the respective information in the table below.

[10 points]

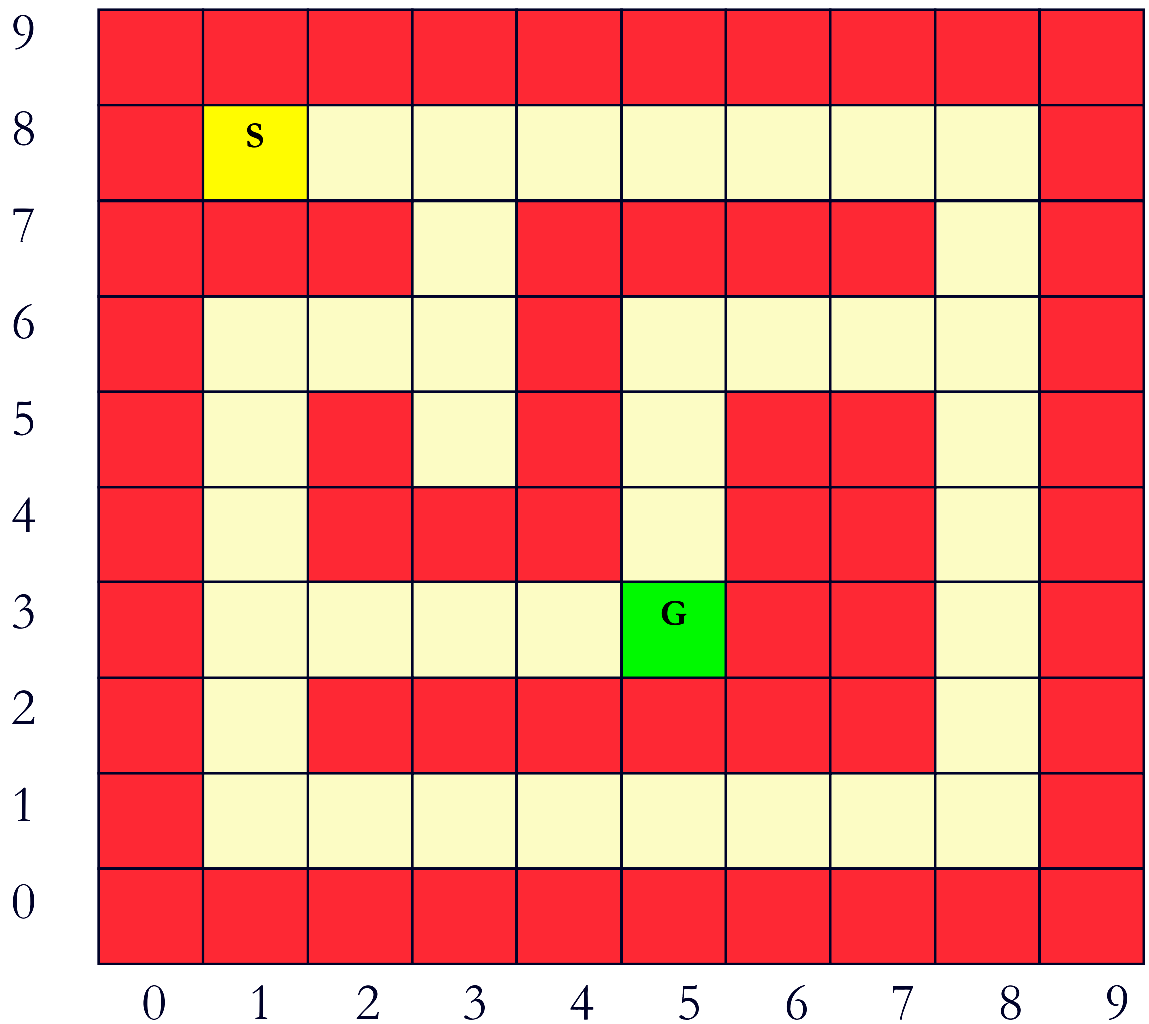


Please note that not all columns may be required for a particular search method, and that the size of the table may not reflect the actual length of the trace. The fringe should be ordered in such a way that the next node to be visited is at the beginning of the queue (in the left-most position). Mark the newly added nodes in each step(e.g. by underlining or circling them), and for each node indicate the value that is used as the ordering criterion in the fringe.

Step	*Current Node*	*Path Cost*	*Heu-ristic*	*F-Cost*	*Fringe (Queue)*
0	**S**	0	8.95	8.95	(**[2,8]**,7.95)
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					

b) Traverse the maze from the starting point **S** to the goal **G** according to the *A* Search* method. For this algorithm, calculate the value of the f-cost for each accessible tile in the maze and write it in the tile, mark the path selected by the agent, and record the respective information in the table below.

[10 points]



Please use the same conventions as in the previous part of this task.

Step	*Current Node*	*Path Cost*	*Heu-ristic*	*F-Cost*	*Fringe (Queue)*
0	**S**	0	8.95	8.95	(**[2,8]**,8.95)
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					
27					
28					
29					
30					

c) Are *local* search algorithms suitable for such a maze traversal problem with the given heuristic?
[5 points]

- ❑ Yes
- ❑ No

Explain your answer.

Total Points: