Last name:_____ First name:_____ SID:_____ Class account login:_____

Collaborators: _____

CS188 Spring 2011 Written 1: Search and CSPs

Due: Monday 2/14, 5:30pm either at the beginning of lecture or in 283 Soda Drop Box. Zero slip time.
Policy: See course webpage.

1 [7pts] All roads lead to Rome

This map shows approximate mean driving times (in hours) between pairs of cities. For each of the following graph search strategies, work out the order in which states are expanded, as well as the path returned by graph search. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal states are Warsaw and Rome, respectively. Remember that in graph search, a state is expanded only once.

[1pt] (a) Depth-first search.

[1pt] (b) Breadth-first search.

[1pt] (c) Uniform cost search.

[1pt] (d) Greedy search with the heuristic: $h(\text{Odesa}) = 20$ hrs, $h(\text{Budapest}) = 12$ hrs, $h(\text{Munich}) = 3$ hrs, $h(\text{Venice}) = 3$ hrs, $h(\text{Rome}) = 0$ hrs, $h(\text{Warsaw}) = 30$ hrs.

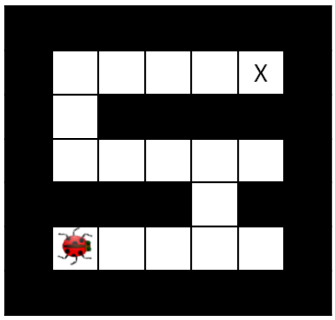
[1pt] (e) $A^\star$ search with the same heuristic.

[2pts] (f) Is the heuristic admissible? Consistent? Briefly justify your answers.

2 Hive Minds (10 points)

You control one or more insects in a rectangular maze-like environment. At each time step, an insect can move into an adjacent square if that square is currently free, or the insect may stay in its current location. In the case of multiple insects adjacent insects cannot swap locations. Squares may be blocked by walls. There are N non-wall squares. Optimality is always in terms of time steps; all actions have cost 1 regardless of the number of insects moving or where they move. For each of the following scenarios, precisely but compactly define the state space and give its size. Then, give a non-trivial admissible heuristic for the problem and the maximum branching factor. Your answers should follow the format in the example case below. Full credit requires a minimal state space (i.e. do not include extra information) and a reasonable non-trivial heuristic. The illustrations are given as examples; your answers should not assume a specific maze.

Example: Lonely Bug You control a single insect as shown in the maze above, which must reach a designated target location X. There are no other insects moving around.



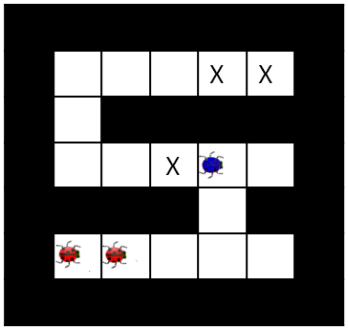
State space description: A tuple (x, y) encoding the x and y coordinates of the insect.

State space size: N.

Heuristic: The Manhattan distance from the insect's location to the target.

Maximum branching factor: 5

(a) (2 pts) Swarm Movement You control K insects, one blue and $K-1$ red. There are K target locations. In each time step all insects move simultaneously (or some may stay in place). Each insect can go to any target location. An example is shown here:



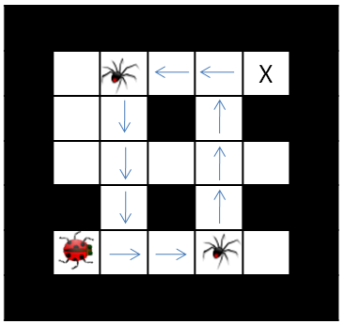
State space description:

State space size:

Maximum branching factor:

Heuristic:

(b) (2 pts) Patrolling Guards You again control a single insect, but there are G spiders patrolling known paths as shown below. Specifically, at time t each guard g will be at position $(x_g(t), y_g(t))$ (in general, guard movements need not be a function of a guard's current location, but you may assume that the tuple of guard positions follows a known pattern that repeats with period T). Similarly to (a), your insect cannot take an action which moves it into either a guard's current location or the location a guard is about to occupy.



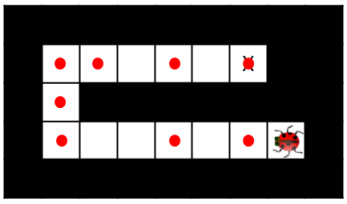
State space description:

State space size:

Maximum branching factor:

Heuristic:

(c) (2 pts) Step on It Your single insect is alone in the maze again. This time, it can speed up as long as it doesn't change direction. Specifically, after a move of v squares in some direction, it can move up to $v + 1$ squares in that same direction on the next time step. It can move fewer than $v + 1$ squares in that direction, as well, and it can move one square in any other direction (or stand still). Moving v squares requires that all intermediate squares passed over, as well as the v-th square, currently be empty. The cost of a multi-square move is still 1 time unit. Let L be the size of the longest straight corridor in the maze. In the example below, $L = 7$ and the dots in the maze below indicate where the insect will be after each time step in the optimal (fewest time step) plan:



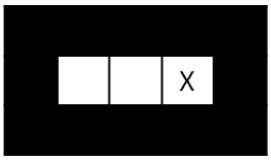
State space description:

State space size:

Maximum branching factor:

Heuristic:

(d) (2 pts) Lost at Night It is night and you control a single insect. You know the maze, but you do not know what square the insect will start in. Let M be the set of non-wall squares in the maze, i.e. $|M| = N$. You must pose a search problem whose solution is an all-purpose sequence of actions such that, after executing those actions, the insect will be on the exit square, regardless of initial position. The insect executes the actions mindlessly and does not know whether its moves succeed: if it uses an action which would move it in a blocked direction, it will stay where it is. For example, in the maze below, moving left twice and then right twice guarantees that the insect will be at the exit regardless of its starting position.



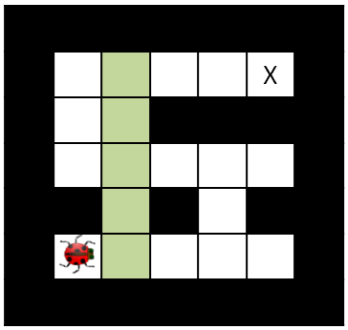
State space description:

State space size:

Heuristic:

Maximum branching factor:

(e) (2 pts) Escape! Your insect is again alone in a maze, but certain squares are filled with pesticide, shown below in green. Those squares are safe to travel through provided the insect does not accumulate more than L time steps in the pesticide, at which point it would die. You must find a solution where no more than L pesticide squares are used.



State space description:

State space size:

Heuristic:

Maximum branching factor:

3 Green Design (7 points)

You are in charge of setting up a greenhouse. There are several plant types: a (C)orpse flower, a (R)ose, a (S)unflower, a (T)ulip, and a (V)ampire plant. There are 5 exhibit slots, numbered 1 to 5, with slot 1 being next to the greenhouse door. You have the following constraints:

(i) No two plants may be in the same square.

(ii) The rose cannot be adjacent to the corpse flower, or its scent will be ruined (the corpse flower feels the same way).

(iii) The vampire plant cannot be adjacent to the rose, sunflower, or tulip, or it will snack on them.

(iv) There must be at least two squares between the vampire plant and the sunflower; the sun burns!

(v) The rose must be closer to the door than the corpse flower, or visitors will refuse to enter the exhibit.

We will formalize this problem as a binary CSP.

(a) (1 pt) Give the constraint on the rose and corpse flower explicitly. Note that this single constraint is spread out between several lines of the description above.

(b) (1 pt) List all pairs of variables (A,B) which have a constraint other than $A \neq B$ constraint and state briefly what the constraints are, e.g. "$(A \neq B - 2) \wedge (A < B)$" . Do not add to the problem constraints that are *consequences* of the above-stated ones (e.g. the CSP does *not* state that the vampire plant must be adjacent to corpse flower even though that is a true consequence of the given constraints).

(c) (1 pt) What will the remaining domains be after arc consistency in enforced? Cross out all filtered values. *Note:* Be careful to only enforce *arc* consistency.

(d) (1 pt) Which variable or variables would be assigned first according to MRV using the domains above?

(e) (2 pts) Assume that we assign $T = 3$ and then enforce arc consistency. What will the remaining domains be? Cross out all filtered values.

(f) (1 pt) List all solutions to this CSP with $T = 3$ or state than none exist.