CS188: Artificial Intelligence, Fall 2011
Written 1: Search and CSPs

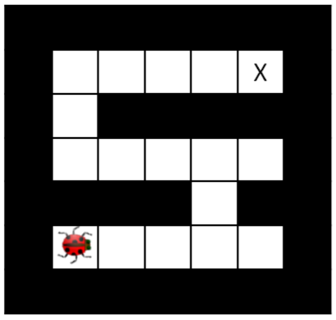
Due: 9/21 submitted **electronically** by 11:59pm (no slip days)
Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually.

Instructions for submitting your assignment can be found on the website via the assignments page:
http://inst.eecs.berkeley.edu/∼cs188/fa11/assignments.html

1 Hive Minds (11 points)

You control one or more insects in a rectangular maze-like environment with dimensions $M \times N$, as shown below. 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. Squares may be blocked by walls, but the map is known. 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, either evaluate the provided heuristics or supply your own, as indicated. Your answers should follow the format of the example case below. Full credit requires a minimal state space (i.e. do not include extra information), but you should answer *for a general instance of the problem*, not simply the map shown.

Example: Lonely Bug You control a single insect as shown in the maze below, 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: $M \cdot N$

Which of the following heuristics are admissible (if any)?

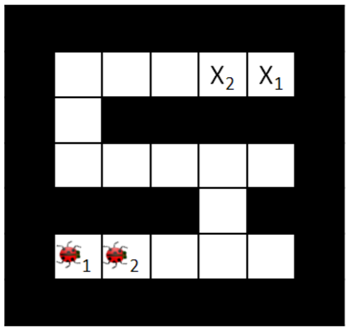
(i) The Manhattan distance from the insect's location to the target.

(ii) The Euclidean distance from the insect's location to the target.

(i) and (ii) are both admissible

(a) (2 pts) Swarm Movement You control K insects, each of which has a specific target ending location X_k. No two insects may occupy the same square. In each time step all insects move simultaneously to a currently free square (or stay in place); adjacent insects cannot swap in a single time step. An example is shown here:

State space description:



State space size:

Which of the following heuristics are admissible (if any)?

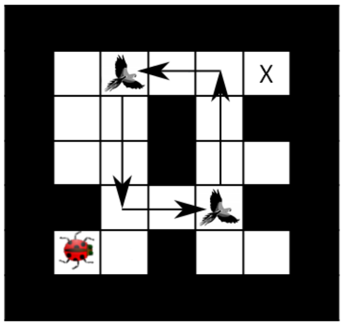
(i) Max of Manhattan distances from each insect's location to its goal.

(ii) Max of costs of optimal paths for each insect to its goal if it were acting alone in the environment, unobstructed by the other insects.

(b) (2 pts) Migrating Birds You again control a single insect, but there are B birds flying along known paths as shown below. Specifically, at time t each bird b will be at position $(x_b(t), y_b(t))$ (in general, bird movements need not be a function of a bird's current location, but you may assume that the tuple of bird positions repeats with period T). Birds might move up to 3 squares per time step.

Your insect *can* share squares with birds and it can even hitch a ride on them! On any time step that your insect shares a square with a bird, the insect may either move as normal *or* move directly to the bird's next location (either action has cost 1, even if the bird travels farther than one square).

State space description:



State space size:

Which of the following heuristics are admissible (if any)?

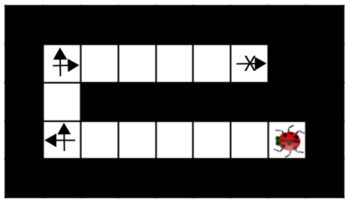
(i) Cost of optimal path to hive in the simpler problem that has no birds.

(ii) Manhattan distance from the current position to the hive

(iii) Manhattan distance from the current position to the hive divided by three.

(c) (2 pts) Jumping Bug Your single insect is alone in the maze again. This time, it has super legs that can take it as far as you want in a straight line in each time step. The disadvantage of these legs is that they make turning slower, so now it takes the insect a time step to change the direction it is facing. Moving v squares requires that all intermediate squares passed over, as well as the vth square, currently be empty. The cost of a multi-square move is still 1 time unit, as is a turning move. As an example, the arrows in the maze below indicate where the insect will be and which direction it is facing after each time step in the optimal (fewest time steps) plan (cost 5):

State space description:



State space size:

Which of the following heuristics are admissible (if any)?

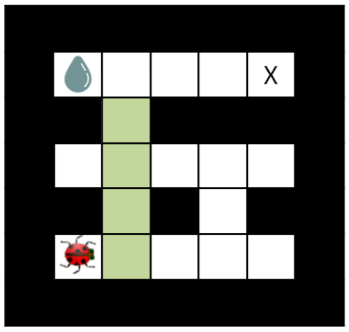
(i) Cost of optimal path to hive in the simpler "lonely bug" problem (with no special jumping ability or turning cost).

(ii) Number of turns in the optimal path in the simpler "lonely bug" problem.

(iii) The number of dimensions (0, 1, or 2) in which the current position differs from the goal.

(d) (2 pts) Clean Your insect is again alone in a maze, but certain squares are filled with poison, shown below in green. If your insect travels through these squares it becomes a risk to the hive. However, if the insect enters a square containing a water droplet, it becomes completely clean again, no matter how much poison it passed through. You must find a solution where the path either never passes through a square containing poison, or if it does, it subsequently passes through a water square before reaching the destination.

State space description:



State space size:

Which of the following heuristics are admissible?

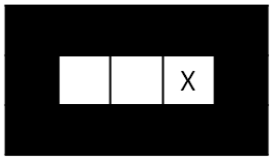
(i) Cost of optimal path to the hive in the relaxed problem which has no poison or water

(ii) Manhattan distance from the current position to the hive

(iii) Manhattan distance to the closest water + Manhattan distance from there to the hive

(e) (3 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. 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:

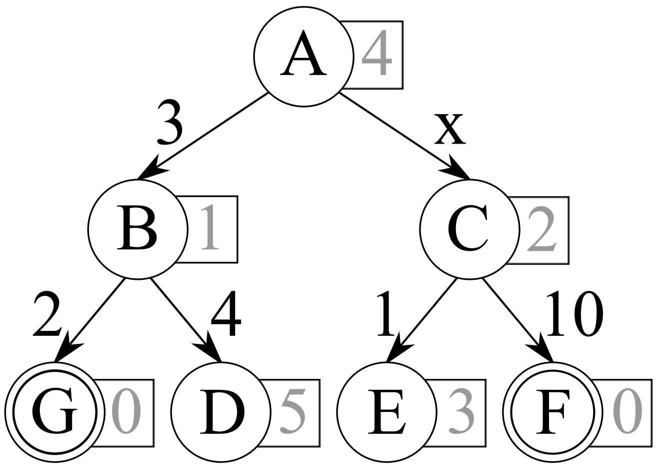


State space size:

Give a heuristic for this problem:

2 Search: Expanded Nodes (10 points)

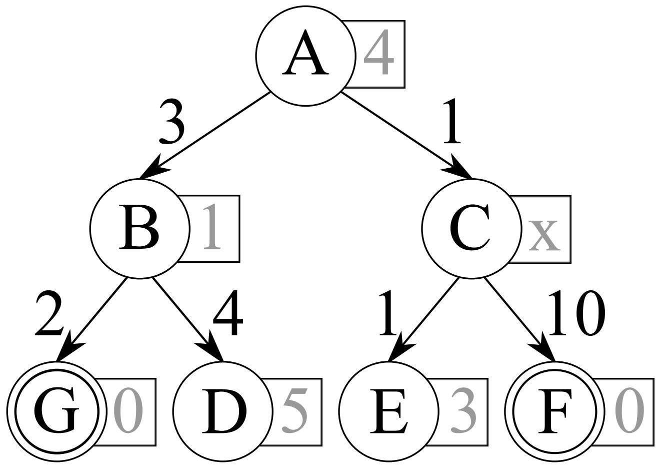
In this question we will consider tree search (i.e. no closed set) on various search problems. First we will consider some simple graphs that contain no cycles:



In this graph, you start at A, and there are two states that pass the goal test: G and F. The numbers next to edges are action costs, and the numbers in boxes are the heuristic values for the nodes they are attached to. Throughout the next five parts, assume ties are broken by expanding the alphabetically earlier node first (so distinctions between $<$ and $\leq$ are meaningful). You may assume that all edge costs are positive and all heuristic values are non-negative.

(a) (1 pt) What range of values for the missing edge weight will lead Uniform Cost search to expand node C?

(b) (1 pt) What range of values for the missing edge weight will lead A* search to expand node C?



Now the missing edge weight has been filled in, but one of the heuristic values is unknown.

(c) (1 pt) What range of numbers for the missing heuristic value will produce an admissible heuristic?

(d) (1 pt) What range of numbers for the missing heuristic value will produce a consistent heuristic?

(e) (1 pt) What range of numbers for the missing heuristic value will lead Greedy search to find a *suboptimal solution*?

Now consider tree search on an arbitrary search problem with max branching factor b. Each search node n has a backward (cumulative) cost of $g(n)$, an admissible heuristic of $h(n)$, and a depth of $d(n)$. Let c be a minimum-cost goal node (with cost $g(c)$), and let s be a shallowest goal node (with depth $d(s)$).

For each of the following, you will give an expression that can be used to determine whether or not a node is in the set of nodes that are expanded before the search terminates. For instance, if we asked for the nodes with positive heuristic value, you would say $h(n) > 0$ to mean the set $\{n : h(n) > 0\}$. Don't worry about ties here (so you won't need to worry about $>$ versus $\geq$). If there are no nodes for which the expression is true, write "false" as the expression.

(f) (1 pt) Give an expression (i.e. an inequality in terms of the above quantities) for nodes n that will be expanded in a breadth-first search.

(g) (1 pt) Give an expression for nodes n that will be expanded in a uniform cost search.

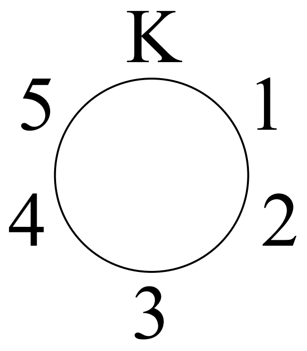
(h) (1 pt) Give an expression for nodes n that will be expanded in an A* search with a consistent heuristic $h(n)$.

(i) (1 pt) Let h_1 and h_2 be two consistent heuristics such that $\forall n, h_1(n) \geq h_2(n)$. Give an expression for nodes n that *will* be expanded in an A* search using h_1 but *not* when using h_2.

(j) (1 pt) Give an expression for nodes n that will be expanded in an A* search using h_2 but will not when using h_1.

3 CSPs: Animal Council (7 points)

You are in charge of planning the next meeting of the Australian animal council. There are several animals being represented: Bogong (M)oths, (D)ingos, (E)chidnas, Frill-necked (L)izards, (G)oannas and (K)angaroos. They will all sit at a round table, with the Kangaroo sitting at the top of the table, as shown in the diagram below.



You will be working out where the other five animals sit, with the following constraints:

(i) No two animals may be in the same seat.

(ii) The frill-necked lizard cannot be next to the bogong moth or the dingo (it might eat the moth, or be attacked by the dingo).

(iii) The echidna must be next to the dingo (mammals unite!).

(iv) The goanna must sit in the position with number one less than the frill-necked lizard's seat (lizards unite... carefully).

(v) The dingo must be closer to the kangaroo than the frill-necked lizard is (there are concerns about the reptiles gaining too much power).

We will formalize this problem as a binary CSP, with the animals as variables and the set of seat numbers as their domains.

(a) (1 pt) Give the constraint on the dingo and the frill-necked lizard *explicitly*. Note that this single constraint is spread out between several lines of the description above. Do not add to the problem constraints that are *consequences* of the above-stated ones (e.g. the CSP does *not* state that the dingo must be adjacent to the kangaroo, even though that is a true consequence of the given constraints). *Hint:* the correct answer is of size 4.

(b) (1 pt) List all pairs of variables (A,B) which have a constraint other than the different(A,B) constraint and state briefly (implicit is ok here) what the constraints are (e.g. "(A, B): different(A,B) $\wedge\neg$adjacent(A,B) $\wedge$ A < B"). As in the previous question, do not add to the problem constraints that are *consequences* of the above-stated ones. You may find it useful to include the constraints between L and D that you worked out in the previous question, but it is not required.

(c) (2 pt) What will the remaining domains be after arc consistency is enforced? Cross out all filtered values. When typing up your solution, list the remaining values. *Note:* Be careful to only enforce *arc consistency*, and don't forget about the $\neq$ constraints.

$$\begin{array}{r}
D = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
E = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
G = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
L = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
M = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ]
\end{array}$$

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

(e) (1 pts) Assume that we assign $L = 3$ and then enforce arc consistency. What will the remaining domains be? Cross out all filtered values. When typing up your solution, list the remaining values.

$$\begin{array}{r}
D = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
E = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
G = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ] \\
L = [\ \ \ 3\ \ \ ] \\
M = [\ 1\ \ 2\ \ 3\ \ 4\ \ 5\ ]
\end{array}$$

(f) (1 pt) List all solutions to this CSP or state that none exist.

4 CSPs: Maze Layout (7 points)

You are tasked with designing a new board layout for a Pacman-style game where each square is either empty or filled with a wall. The layout rules are simple: (i) the border of the board must consist of walls and (ii) the board cannot have dense open spaces or dense walls, defined as a 2x2 block of the same value. The entire board is $M \times N$. Assume there is a variable $S_{ij} \in \{e, w\}$ for each square (i, j), whose value indicates whether that square is *empty* or filled with a *wall*.

(a) (1 pt) State the constraints of the problem (implicit statements are ok, but be precise). Don't forget about unary constraints!

(b) (2 pts) Imagine that your CSP solver only handles unary and binary constraints. Come up with an alternate formulation which adds a new variable Q_{ij} for each 2-by-2 block with upper left corner at (i, j). This new formulation should use only unary and/or binary constraints (though the new variable's domains need not be binary). State precisely what the variables, domains, and constraints are for your formulation.

Note: This question has been updated to clarify the use of unary constraints.

(c) (2 pts) Describe briefly but precisely how any n-ary (i.e. more than binary) CSP can be converted into a binary one through the introduction of new variables.

(d) (2 pts) You notice that the resulting boards do not look quite right. You want to add a new requirement: every free square must be reachable from every other one. You can add an extra variable, *reachability*, to capture whether this constraint is satisfied, but then you need to know whether *reachability* is true or false.

Assume you have the original non-binary formulation in (a) and that you have a partial map, where every square is either empty, filled by a wall, or currently undecided. How can you determine the value of the *reachability* variable efficiently?