Written HW1

INSTRUCTIONS

- **Written HW Due:** Friday 6/28/2019 at 11:59 PM (submit via Gradescope).
- **Corresponding self-grade due:** Tuesday 7/2/2019 at 11:59 PM (submit via Gradescope).
- **Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually.
- **Submissions:** Your submission should be a PDF that matches this template. Each page of the PDF should align with the corresponding page of the template (page 1 has name/collaborators, question 1 begins on page 2, etc.). Do not reorder, split, combine, or add extra pages. The intention is that you print out the template, write on the page in pen/pencil, and then scan or take pictures of the pages to make your submission. You may also fill out this template digitally (e.g. using a tablet.)

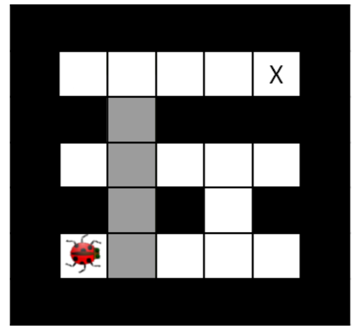
  Don't forget to fill up the self-grade assignment corresponding to this homework on Gradescope after the deadline of this assignment when the solution is released. You will have until the due date of the next homework to complete your self-grade. For more information, please visit the homework policy post on Piazza.

First Name	
Last Name	
SID	
Collaborators	

For staff use only

Q. 1	Q. 2	Total
/15	/15	/30

1. (15 points) Question Menagerie



Parts (a) and (b) The hive of insects needs your help again. As before, you control an insect in a rectangular maze-like environment with dimensions $M \times N$, as shown to the right. At each time step, the insect can move into a free adjacent square or stay in its current location. All actions have cost 1.

In this particular case, the insect must pass through a series of partially flooded tunnels. Flooded squares are lightly shaded in the example map shown. The insect can hold its breath for A time steps in a row. Moving into a flooded square requires your insect to expend 1 unit of air, while moving into a free square refills its air supply.

(a) (4 pt) Give a minimal state space for this problem (i.e. do not include extra information). You should answer for a general instance of the problem, not the specific map shown.

A tuple of location coordinates $m \in \{1, \dots, M\}$ and $n \in \{1, \dots, N\}$ and the remaining air supply $a \in \{1, \dots, A\}$.

(b) (4 pt) Give the size of your state space.

$M \times N \times A$

Parts (c), (d), and (e) Consider a search problem where all edges have cost 1 and the optimal solution has cost C. Let h be a heuristic which is $\max\{h^* - k, 0\}$, where h^* is the actual cost to the closest goal and k is a nonnegative constant.

(c) (3 pt) Which of the following statements are true?

(i) h is admissible.

(ii) h is consistent.

(iii) A* tree search (no closed list) with h will be optimal.

(iv) A* graph search (with closed list) with h will be optimal.

(i): h* is admissible and h $\leq h^*$.
(ii): h^* is consistent and subtracting a constant from both sides does not change the underlying inequality.
(iii): Tree search requires admissibility.
(iv): Graph search requires admissibility and consistency.

(d) (2 pt) Which of the following is the most reasonable description of how much more work will be done (= how many more nodes will be expanded) with heuristic h compared to h^*, as a function of k?

(i) Constant in k

(ii) Linear in k

(iii) Exponential in k

(iv) Unbounded

At $k = 0$, only the d nodes on an optimal path to the closest goal are expanded for search depth (= optimal path length) d. At $k = \max(h)$, the problem reduces to uninformed search and BFS expands b^d nodes for branching factor b . In general, all nodes within distance k of the closest goal will have heuristic $h = 0$ and uninformed search may expand them. Note that search reduces to BFS since A^* with $h = 0$ is UCS and in this search problem all edges have cost 1 so path cost = path length.

Now consider the same search problem, but with a heuristic h' which is 0 at all states that lie along an optimal path to a goal and h^* elsewhere.

(e) (3 pt) Which of the following statements are true?

(i) h' is admissible.

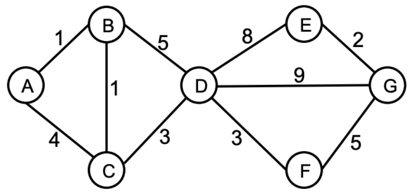
(ii) h' is consistent.

(iii) A* tree search (no closed list) with h' will be optimal.

(iv) A* graph search (with closed list) with h' will be optimal.

h' is not consistent as needed for optimality of graph search. Consistency is violated by any edge connecting a state outside of the optimal path to the optimal path since h' drops faster than the reduction in the true cost.

2. (15 points) Search



Node	h_1	h_2
A	9.5	10
B	9	12
C	8	10
D	7	8
E	1.5	1
F	4	4.5
G	0	0

Consider the state space graph shown above. A is the start state and G is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions. Note that the heuristic h_1 is consistent but the heuristic h_2 is not consistent.

(a) [6 pts] **Possible paths returned** For each of the following graph search strategies (*do not answer for tree search*), mark which, if any, of the listed paths it could return. Note that for some search strategies the specific path returned might depend on tie-breaking behavior. In any such cases, make sure to mark *all* paths that could be returned under some tie-breaking scheme.

Search Algorithm	**A-B-D-G**	**A-C-D-G**	**A-B-C-D-F-G**
Depth first search	(i)	(ii)	(iii)
Breadth first search	(iv)	(v)	(vi)
Uniform cost search	(vii)	(viii)	(ix)
A* search with heuristic h_1	(x)	(xi)	(xii)
A* search with heuristic h_2	(xiii)	(xiv)	(xv)

The return paths depend on tie-breaking behaviors so any possible path has to be marked. DFS can return any path. BFS will return all the shallowest paths, i.e. A-B-D-G and A-C-D-G. A-B-C-D-F-G is the optimal path for this problem, so that UCS and A* using consistent heuristic h_1 will return that path. Although h_2 is not consistent, it will also return this path.

(b) [9 pts] **Heuristic function properties**
Suppose you are completing the new heuristic function h_3 shown below. All the values are fixed except $h_3(B)$.

Node	A	B	C	D	E	F	G
h_3	10	?	9	7	1.5	4.5	0

For each of the following conditions, write the set of values that are possible for $h_3(B)$. For example, to denote all non-negative numbers, write $[0, \infty]$, to denote the empty set, write $\varnothing$, and so on.

(i) [2 pts] **What values of $h_3(B)$ make h_3 admissible?**

To make h_3 admissible, $h_3(B)$ has to be less than or equal to the actual optimal cost from B to goal G, which is the cost of path $B - C - D - F - G$, i.e.12.
The answer is $0 \leq h_3(B) \leq 12$

(ii) [3 pts] **What values of $h_3(B)$ make h_3 consistent?**

$h(A) \leq c(A, B) + h(B) \quad h(B) \leq c(B, A) + h(A)$
$h(C) \leq c(C, B) + h(B) \quad h(B) \leq c(B, C) + h(C)$
$h(D) \leq c(D, B) + h(B) \quad h(B) \leq c(B, D) + h(D)$
Filling in the numbers shows this results in the condition: $9 \leq h_3(B) \leq 10$

(iii) [4 pts] **What values of $h_3(B)$ will cause A* graph search to expand node A, then node C, then node B, then node D in order?**

In order to make A^* graph search expand node A, then node C, then node B, suppose $h_3(B) = x$, we need

$1 + x > 13$

$5 + x < 14$ (expand B') or $1 + x < 14$ (expand B)

so we can get $12 < h_3(B) < 13$