CS 188 Fall 2014

Introduction to Artificial Intelligence

HW2

INSTRUCTIONS

- **Due:** Wednesday, September 17, 2014 11:59 PM
- **Policy:** Can be solved in groups (acknowledge collaborators) but must be written up individually. However, we strongly encourage you to first work alone for about 30 minutes total in order to simulate an exam environment. Late homework will not be accepted.
- **Format:** Submit the answer sheet pdf containing your answers. You should solve the questions on this handout (either through a pdf annotator, or by printing, then scanning; we recommend the latter to match exam setting). **Make sure that your answers (typed or handwritten) are within the dedicated regions for each question/part. If you do not follow this format, we may deduct points.**
- **How to submit:** Go to www.pandagrader.com. Log in and click on the class CS188 Fall 2014. Click on the submission titled HW 2 and upload your pdf containing your answers. If this is your first time using pandagrader, you will have to set your password before logging in the first time. To do so, click on "Forgot your password" on the login page, and enter your email address on file with the registrar's office (usually your @berkeley.edu email address). You will then receive an email with a link to reset your password.

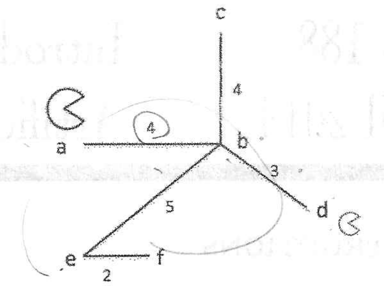
Last Name	Lee
First Name	Charles
SID	23728976
Email	charleslee94@berkeley.edu
Collaborators	Daniel He

For staff use only

Q. 1	Q. 2	Q. 3	Q. 4	Total
/36	/50	/20	/44	/150

1. **(36 points) Rendezvous**

Pacman and Pacbaby are trying to reach each other on a 2D map, as shown to the right. At each turn Pacman and Pacbaby must run from one rest spot to another (labeled a, b, ...). The amount of time required to get to each rest spot from an adjacent rest spot is equal to the distance between the two, $d(i,j)$, shown at right. Whichever Pac arrives first must wait for the other one before the next turn begins. Pacman and Pacbaby are trying to meet at the same rest spot as quickly as possible.



(a) **(6 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.

Put your answer to 1a here:

n = # rest spots

2n

(b) **(18 pt)** Let Adjacent(i) be the set of rest spots adjacent to rest spot i. Define a transition model, goal test and step cost function for this problem.

Put your answer to 1b here:

Transition

Result(Adj(a), (P_m, P_b)) = b

Result(Adj(b), (P_m, P_b)) = a, c, d, e

Result(Adj(c), (P_m, P_b)) = b

Result(Adj(d), (P_m, P_b)) = b

Result(Adj(e), (P_m, P_b)) = b, f

Result(Adj(f), (P_m, P_b)) = e

Goal Test = $(P_m == P_b)$

Stepcost: Σ(distance of Pacman) + Σ(distance of Pacbaby)

(c) **(12 pt)** Write a nontrivial admissible heuristic for this problem and prove its admissibility.

Put your answer(s) to 1c here:

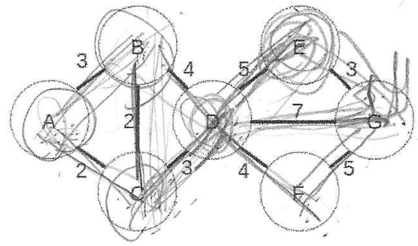
h(n) = Average distance btwn point of man and point of baby)

Proof: Average of two distances will always be less than the sum of the two.

2. (50 points) Search

A

AB AC



AC

Node	h_1	h_2	
A	12.5	11	12
B	12	10.5	11
C	11	9	10
D	6	6.75	7
E	1	2	3
F	4.5	4.5	5
G	0	0	9

12

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.

(a) [4 pts] **Admissibility and consistency part I**

(i) [2 pts] For heuristics h_1 and h_2, determine whether each is admissible.

h_1 is not admissible, h_2 is admissible

(ii) [2 pts] For heuristics h_1 and h_2, determine whether each is consistent.

h_1 is not consistent, h_2 is consistent

(b) [10 pts] **Order of expansion**
List out the order of node expansion for each of the graph search strategies.

(i) [2 pts] Depth First Search

ABCDEG

(ii) [2 pts] Breadth First Search

ABCDEFG

(iii) [2 pts] Uniform Cost Search

ACBDFEG

(iv) [2 pts] A* Search with heuristic h_1

ACDEG

(v) [2 pts] A* Search with heuristic h_2

ACDG

(c) [12 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)

(d) [18 pts] **Heuristic function properties**

Suppose you are completing the new heuristic function h_3 shown below. All the values are fixed except $h_3(\cancel{B})$.

C

Node	A	B	C	D	E	F	G
h_3	11	10	?	6	2.5	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) [4 pts] **What values of $h_3(C)$ make h_3 admissible?**

Put your answer to 2d(i) here:

$h(C)$ [9, 10]

(ii) [6 pts] **What values of $h_3(C)$ make h_3 consistent?**

Put your answer to 2d(ii) here

9

(iii) [8 pts] **What values of $h_3(C)$ will cause A* graph search to expand node A, then node C, then node B, then node D in order?**

Put your answer to 2d(iii) here

$\varnothing$

(e) [6 pts] **Admissibility and consistency part II**

Let h_4 and h_5 be admissible heuristics. Determine whether each of the following is necessarily admissible.

(i) [2 pts] [necessarily admissible / not necessarily admissible] $\max(h_4, h_5)$

(ii) [2 pts] [necessarily admissible / not necessarily admissible] $\min(h_4, h_5)$

(iii) [2 pts] [necessarily admissible / not necessarily admissible] $(h_4 + h_5)/2$

3

3. **(20 points) Local Search**

Give the name of the algorithm that results from each of the following special cases:

(a) Local beam search with k=1

Put your answer(s) to 3a here: generates 1 start state, hill climbing
replace start w/ best state (neighbor)

(b) Local beam search with one initial state and no limit on the number of states retained.

Put your answer(s) to 3b here: start w/ 1 state, generate all successors,
then generate all successors of all those successors,
breadth first search

(c) Simulated annealing with T=0 at all times (and omitting the termination test).

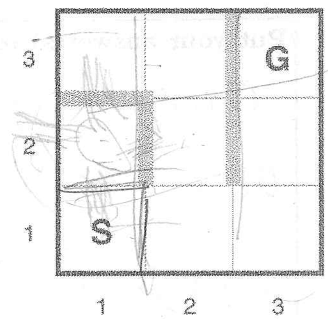
Put your answer(s) to 3c here: T=0 means small chance for "bad move"
hill climbing.

(d) Simulated annealing with T=∞ at all times.

Put your answer(s) to 3d here: T=∞ means infinite "bad moves"
this is a random walk

4. (44 points) Search Agents and Partial Observability

Pacman is trapped in a 3 × 3 maze like the one shown to the right. Pacman starts at (1,1), the goal is at (3,3), and the actions Up, Down, Left, Right have their usual effects unless blocked by a wall. Pacman does *not* know where the internal walls are. In any given state, Pacman can see the set of legal actions.
This problem can be viewed as an offline search problem in belief-state space, where the initial belief state includes all possible environment configurations.



(a) How large is the initial belief state?

Put your answer(s) to 4a here:

1

(b) How large is the space of belief states?

Put your answer(s) to 4b here:

$9 \cdot 2^{12}$

(c) How many distinct percepts are possible in the initial state?

Put your answer(s) to 4c here:

4

(d) Describe in words the transition model for this offline search problem.

Put your answer(s) to 4d here:

Check if action is legal, if so, perform action. if action is not legal change action (up, down, left, right)

(e) How large (roughly) is the complete contingency plan for this problem? Explain how you arrived at this number.

Put your answer(s) to 4e here:

mn^4

where m is width of board, n is length.