HW1: Written

Friday, March 15, 2019 11:17 AM

CS_188_Fa...

CS 188
Fall 2018

Introduction to
Artificial Intelligence

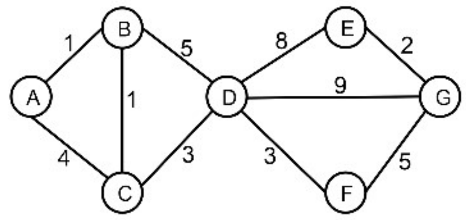
Written HW 1

Due: Tuesday 9/4/2018 at 11:59pm (submit via Gradescope)

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

First name	
Last name	
SID	
Collaborators	

Q1. 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) 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			
Breadth first search			
Uniform cost search			
A* search with heuristic h_1			
A* search with heuristic h_2			

PATH, NOT EXPLORED

(b) 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) What values of $h_3(B)$ make h_3 admissible?

[0, 12]

H(b) <= delta(B, G)

(ii) What values of $h_3(B)$ make h_3 consistent?

10 - 1 = 9 <= H(b) <= h(a) + 1 = 10+1 = 11

[9, 10]

H(v) - w(v, u) <=H(u) <= w(u, v) + h(v)

H(u) <= Delta(u, v) <= w(u, v) + h(v)

9 -1 = 8 <= H(b) <= H(c) +1 = 1 9+1 = 10

7 - 5 = 2 <= H(b) <= H(d) + 5 = 7+5 = 12

(iii) 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?

A, C, B, D Not possible. B is always preferred

A heuristic is logical (eats up obvious slack)
If the heuristic for u is less than
The cost to get to another node + the heuristic to get to v

H(b) must be at least 9, because if we travel frm a to b,
We know that the heuristic of b + 1 must upper bound h(a)
If it's not, h(a) has some unnecessary slack we should pull out.

Q2. n-pacmen search

Consider the problem of controlling n pacmen simultaneously. Several pacmen can be in the same square at the same time, and at each time step, each pacman moves by at most one unit vertically or horizontally (in other words, a pacman can stop, and also several pacmen can move simultaneously). The goal of the game is to have all the pacmen be at the same square in the minimum number of time steps. In this question, use the following notation: let M denote the number of squares in the maze that are not walls (i.e. the number of squares where pacmen can go); n the number of pacmen; and $p_i = (x_i, y_i) : i = 1 \dots n$, the position of pacman i. Assume that the maze is connected.

(a) What is the state space of this problem?

M is number of valid positions
N is the number of pacmen
Pacmen position can be described mapped to any of the m positions
{ (p_0, p_1, …, p_n) }, where p_i in {1, … m}

(b) What is the size of the state space (not a bound, the exact size)?

|S| = m ** n

(c) Give the tightest upper bound on the branching factor of this problem.

Any pacman can have up to 5 transitions (one is a self-transition)
Therefore the number of possible successors of a given state is 5**n

(d) Bound the number of nodes expanded by uniform cost *tree* search on this problem, as a function of n and M. Justify your answer.

B = (5**n)**m

2

You can do one better

We can bound solution max depth by M/2, the furthest pacmen can be away is
Each other they can meet at M/2

(e) Which of the following heuristics are admissible? Which one(s), if any, are consistent? Circle the corresponding Roman numerals and briefly justify all your answers.

1. The number of (ordered) pairs (i, j) of pacmen with different coordinates:

   $h_1(p_1, \dots, p_n) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \mathbf{1}[p_i \neq p_j]$ where $\mathbf{1}[p_i \neq p_j] = \begin{cases} 1 & \text{if } p_i \neq p_j \\ 0 & \text{otherwise} \end{cases}$

   (i) Consistent? (ii) Admissible?

   Not admissible. Counter-example. 4 pacmen can be all adjacent to one square. The heuristic would be 6, but we can find a "path" of length 1 to put all the pacmen on the same square.
   All transitions are cost 1. h(v) - 1 <= h(u) <= h(v) + 1. Not consistent either by the same counterexample.

2. $h_2((x_1, y_1), \dots, (x_n, y_n)) = \frac{1}{2} \max \{\max_{i,j} |x_i - x_j|, \ \max_{i,j} |y_i - y_j|\}$

   (i) Consistent? (ii) Admissible?

   Admissible, for the two furthest pacmen in x to get to each other, they have to move at LEAST 1/2 (xj - xi). For the two furthest pacmen in y to get to each other, they have to move at LEAST 1/2 (yj - yi)

   The half factor is both of them moving towards each other each step.

   H(v) - 1 <= H(u) <= h(v) + 1

   NOT consistent. Counterexample, consider 4 pacmen. All on the same location. In 1 step, the 4 pacmen can move to all unique positions, making the heuristic value escalaate from 0 to 1/2*(2+2) = 2

   This exceeds the bounds required for conssitency s M, moving towards