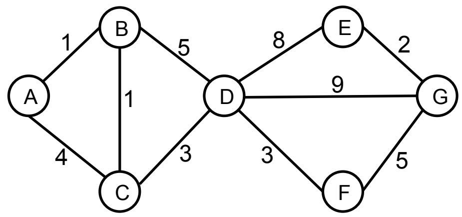
CS 188 Fall 2018 | Introduction to Artificial Intelligence | Written HW 1

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	Jing
Last name	Tan
SID	3034428726
Collaborators	None.

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			√

(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?

$0 \le h_3(B) \le 12$

From B to Goal, the shortest path has a total cost of $(1+3+3+5) = 12$

According to the property of heuristic function h_3 is the underestimate of the actual cost to goal, thus $h_3(B) \le 12$

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

According to the definition of consistency of Heuristics, we have

$-cost(AB) \le h_3(B) - h_3(A) \le cost(BA)$ $\Rightarrow$ $-1 \le h_3(B) - 10 \le 1$ $\Rightarrow$ $9 \le h_3(B) \le 11$

$-cost(CB) \le h_3(B) - h_3(C) \le cost(BC)$ $\Rightarrow$ $-1 \le h_3(B) - 9 \le 1$ $\Rightarrow$ $8 \le h_3(B) \le 10$ $\Rightarrow$ $9 \le h_3(B) \le 10$.

$-cost(DB) \le h_3(B) - h_3(D) \le cost(BD)$ $-5 \le h_3(B) - 7 \le 5$ $2 \le h_3(B) \le 12$

$\therefore$ $h_3(B)$ can be any value between 9 and 10.

(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?

Any $h_3(B) \in (12, 13)$ will do.

To get C expanded before B, we have $(1 + h_3(B)) > 4 + 9 = 13$

To get B expanded before D, and $(4 + 1 + h_3(B)) > 1 + h_3(B)$,

we have $(1 + h_3(B)) < 4 + 3 + 7 = 14$

$\therefore$ $12 < h_3(B) < 13$

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 $p_i = (x_i, y_i) : i = 1 \ldots n$, the position of pacman i. Assume that the maze is connected.

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

The state space is the possible position $p_i = (x_i, y_i)$ for each pacman i.

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

The size of the state space is M^n,
M is the number of possible position of a pacman
n is the number of Pacmans.

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

For every pacman, at each time step there're at most 5 actions can be adopted. Thus, for each state, there are 5^n possible successor state. Therefore, the branching factor upper bound is 5^n.

In my opinion, the bound given by adding all nodes from every layer is more tight and accurate, which is $\frac{5^{\frac{nM}{2}+1}-1}{5^n-1}$

(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.

BFS expanded nodes are (loosely) bounded by the nodes in the deepe which b^D, b being the branching factor and D being the max depth, $\frac{1}{2}M$

Since the cost is the time consumes, and one step at a time for each pacman, uniform Tree search is no different than breadth-first search. The number of nodes expanded is at most $\sum_{i=0}^{M/2} (5^n)^i = 1\frac{1-5^{(nM/2+1)}}{1-5^n} = \frac{5^{(nM/2+1)}-1}{5^n-1}$.
$M/2$ being the depth, for pacmen move simultaneously, depth shall be considered to reduce by half.

$5^{nM/2}$

(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:

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

(i) Consistent? (ii) Admissible?

h_1 is neither consistent nor admissable.
Here is the counterexample: $p_1(0,1)$, $p_2(0,-1)$, $p_3(1,0)$, $p_4(-1,0)$. According to the definition of h_1, we have $h_1(p_1,p_2,p_3,p_4) = 6$. However, if p_1 moves down a step, p_2 moves up a step, p_3 moves left and p_4 moves right. They all move in one timestep; to be in the same space $(0,0)$. $h_1 = 4 >$ real cost $= 1$.
Thus, not admissable and consistent.

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

(i) Consistent? (ii) Admissible?

If we draw a circle as small as possible to cover all pacmen in or on it, the radius of the circle is approximately $\frac{1}{2}\max\{\max_{i,j}|x_i-x_j|, \max_{i,j}|y_i-y_j|\}$,
The fastest way for pacmen to be in same square is go somewhere near each one of them, which is the center of this circle. In addition, pacmen can only move vertially or horizontally, thus radius is stiller shorter (longer than) than the actual longest path to center. If at least one pacman moves, we have $\Delta h_2 \leq$ cost. If the pacman is not on the circle, moving it won't change h_2, $\Delta h_2 = 0 \leq$ cost
If the pacman is on the circle, then as said above, $\Delta h_2 \leq \Delta r \leq \Delta$path $=$ cost. Thus, h_2 is consistent.