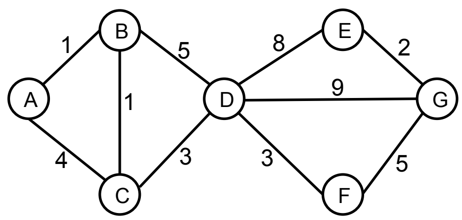
CS 188 Spring 2019 | Introduction to Artificial Intelligence | # Written HW 1 Sol.

Self-assessment due: Monday 2/11/2018 at 11:59pm (submit via Gradescope)

For the self assessment, **fill in the self assessment boxes in your original submission** (you can download a PDF copy of your submission from Gradescope). For each subpart where your original answer was correct, write "correct." Otherwise, write and explain the correct answer.

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	x	x	x
Breadth first search	x	x	
Uniform cost search			x
A* search with heuristic h_1			x
A* search with heuristic h_2			x

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) 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?

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

All the other nodes except node B satisfy the consistency conditions. The consistency conditions that do involve the state B are:

$$h(A) \leq c(A,B) + h(B) \qquad h(B) \leq c(B,A) + h(A)$$
$$h(C) \leq c(C,B) + h(B) \qquad h(B) \leq c(B,C) + h(C)$$
$$h(D) \leq c(D,B) + h(B) \qquad h(B) \leq c(B,D) + h(D)$$

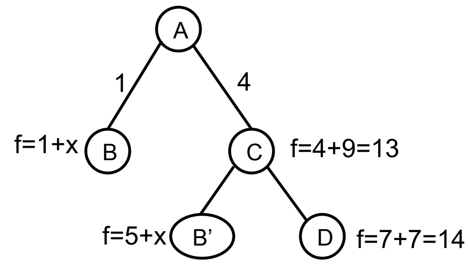
Filling in the numbers shows this results in the condition: $9 \leq h_3(B) \leq 10$

(iii) 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?

The A* search tree using heuristic h_3 is on the right. In order to make A* graph search expand node A, then node C, then node B, suppose $h_3(B) = x$, we need

$$1 + x > 13$$
$$5 + x < 14 \quad (expand\ B') \quad or \quad 1 + x < 14 \quad (expand\ B)$$

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



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

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

n-Tuples, where each entry is in $\{1, \dots, M\}$.

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

M^n

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

5^n (Each pacman has five actions: Stop and the 4 directions).

(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.

As in breadth-first search, the number of nodes expanded is bounded by b^D, with b being the branching factor, and D being the maximum depth of the search tree. Therefore, the answer is $5^{\frac{nM}{2}}$, because the max depth of a solution is $M/2$ while the branching factor is 5^n. How do we know the max depth of a solution? Imagine the worst possible case: bringing together pacmen that are as far away from each other as possible. Since there are M total navigable cells, the maximum number of moves to accomplish this is $M/2$.

(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? Neither. Consider $n = 3$, no wall, and state s such that pacmen are at positions $(i+1, j), (i-1, j), (i, j+1)$. All pacmen can meet in one step, but $h(s) > 1$.

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: imagine a relaxed problem where there are no walls and pacmen can move diagonally. The number of steps needed to solve that relaxed problem is ceil $\frac{1}{2} \max(\max_{i,j} |x_i - x_j|, \max_{i,j} |y_i - y_j|)$. Therefore ceil h_2 is admissible. So, h_2 is also admissible, because $h_2 <$ ceil h_2. It is also consistent because each absolute value will change by at most 2 per step, meaning that h_2 will decrease by at most 1 for each action (actions have cost 1).