CS 188
Fall 2021

Midterm Review Search

Q1. They See Me Rolling (Search Problem)

Pacman buys a car to start Rolling in the Pac-City! But driving a car requires a new set of controls because he can now travel faster than 1 grid per turn (second). Instead of solely moving [North, South, East, West, Stop], Pacman's car has two distinct integers to control: throttle, and steering.

Throttle: $t_i \in \{1, 0, -1\}$, corresponding to {Gas, Coast, Brake}. This controls the **speed** of the car by determining its acceleration. The integer chosen here will be added to his velocity for the next state. For example, if Pacman is currently driving at 5 grid/s and chooses Gas he will be traveling at 6 grid/s in the next turn.
Steering: $s_i \in \{1, 0, -1\}$, corresponding to {Turn Left, Neutral, Turn Right}. This controls the **direction** of the car. For example, if he is facing North and chooses Turn Left he will be facing West in the next turn.

(a) Suppose Pac-city has dimension m by n, but only $k < mn$ squares are legal roads. The speed limit of Pac-city is 3 grid/s. For this sub-part only, suppose Pacman is a law-abiding citizen, so $0 \leq v \leq 3$ at all time, and he only drives on legal roads.

(i) Without any additional information, what is the **tightest upper bound** on the size of state space, if he wants to search a route (not necessarily the shortest) from his current location to anywhere in the city. Please note that your state space representation must be able to represent **all** states in the search space.

◯ $4mn$ ◯ $4k$ ◯ $12mn$ ◯ $12k$ ◯ $16mn$ ◯ $16k$ ◯ $48mn$ ◯ $48k$

(ii) What is the maximum branching factor? The answer should be an integer.

(iii) Which algorithm(s) is/are guaranteed to return a path between two points, if one exists?

☐ Depth First Tree Search ☐ Breadth First Tree Search
☐ Depth First Graph Search ☐ Breadth First Graph Search

(iv) Is Breadth First Graph Search guaranteed to return the path with the shortest grid distance?

◯ Yes ◯ No

(b) Now let's remove the constraint that Pacman follows the speed limit. Now Pacman's speed is limited by the mechanical constraints of the car, which is 6 grid/s, double the speed limit.

Pacman is now able to drive twice as fast on the route to his destination. How do the following properties of the search problem change as a result of being able to drive twice as fast?

(i) Size of State Space:

◯ Increases ◯ Stays the same ◯ Decreases

(ii) Maximum Branching Factor:

◯ Increases ◯ Stays the same ◯ Decreases

For the following part, **mark all choices that could happen on any graph**

(iii) The number of nodes expanded with **Depth** First Graph Search:

☐ Increases ☐ Stays the same ☐ Decreases

(c) Now we need to consider that there are $p > 0$ police cars waiting at $p > 0$ distinct locations trying to catch Pacman riding dirty!! All police cars are stationary, but once Pacman takes an action which lands him in the same grid as one police car, Pacman will be arrested and the game ends.

Pacman wants to find a route to his destination, without being arrested. How do the following properties of the search problem change as a result of avoiding the police?

(i) Size of State Space:
◯ Increases ◯ Stays the same ◯ Decreases

(ii) Maximum Branching Factor:
◯ Increases ◯ Stays the same ◯ Decreases

For the following part, **mark all choices that could happen on any graph**

(iii) Number of nodes expanded with **Breadth** First Graph Search:
☐ Increases ☐ Stays the same ☐ Decreases

Q2. Search

Each True/False question is worth 1 points. Leaving a question blank is worth 0 points. **Answering incorrectly is worth −1 points.**

(a) Consider a graph search problem where for every action, the cost is at least ϵ, with $\epsilon > 0$. Assume the used heuristic is consistent.

(i) [*true* or *false*] Depth-first graph search is guaranteed to return an optimal solution.

(ii) [*true* or *false*] Breadth-first graph search is guaranteed to return an optimal solution.

(iii) [*true* or *false*] Uniform-cost graph search is guaranteed to return an optimal solution.

(iv) [*true* or *false*] Greedy graph search is guaranteed to return an optimal solution.

(v) [*true* or *false*] A* graph search is guaranteed to return an optimal solution.

(vi) [*true* or *false*] A* graph search is guaranteed to expand no more nodes than depth-first graph search.

(vii) [*true* or *false*] A* graph search is guaranteed to expand no more nodes than uniform-cost graph search.

(b) Let $h_1(s)$ be an admissible A* heuristic. Let $h_2(s) = 2h_1(s)$. Then:

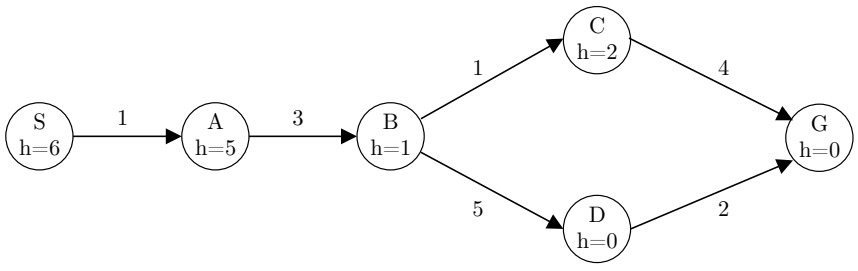
(i) [*true* or *false*] The solution found by A* tree search with h_2 is guaranteed to be an optimal solution.

(ii) [*true* or *false*] The solution found by A* tree search with h_2 is guaranteed to have a cost at most twice as much as the optimal path.

(iii) [*true* or *false*] The solution found by A* graph search with h_2 is guaranteed to be an optimal solution.

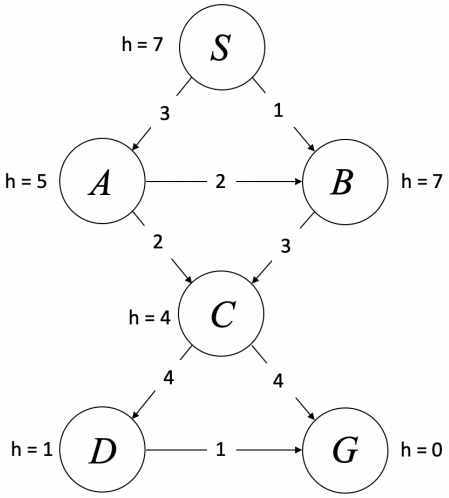
(c) The heuristic values for the graph below are not correct. For which single state (S, A, B, C, D, or G) could you change the heuristic value to make everything admissible and consistent? What range of values are possible to make this correction?

State: [ ] Range: [ ]



Q3. Search Algorithms Potpourri

(a) We will investigate various search algorithms for the following graph. Edges are labeled with their costs, and heuristic values h for states are labeled next to the states. S is the start state, and G is the goal state. In all search algorithms, assume ties are broken in alphabetical order.



(i) Select all boxes that describe the given heuristic values.

☐ admissible ☐ consistent ☐ Neither

(ii) Given the above heuristics, what is the order that the states are going to be expanded in, assuming we run A* graph search with the heuristic values provided.

Index	1	2	3	4	5	Not Expanded
S	◯	◯	◯	◯	◯	◯
A	◯	◯	◯	◯	◯	◯
B	◯	◯	◯	◯	◯	◯
C	◯	◯	◯	◯	◯	◯
D	◯	◯	◯	◯	◯	◯
G	◯	◯	◯	◯	◯	◯

(iii) Assuming we run A* graph search with the heuristic values provided, what path is returned?

◯ $S \to A \to B \to C \to D \to G$ ◯ $S \to A \to C \to G$ ◯ $S \to A \to C \to D \to G$
◯ $S \to B \to C \to G$ ◯ $S \to A \to C \to D \to G$ ◯ $S \to A \to C \to D \to G$
◯ $S \to A \to B \to C \to G$ ◯ None of the above

(iv) Given the above heuristics, what is the order that the states are going to be expanded in, assuming we run greedy graph search with the heuristic values provided.

Index	1	2	3	4	5	Not Expanded
S	◯	◯	◯	◯	◯	◯
A	◯	◯	◯	◯	◯	◯
B	◯	◯	◯	◯	◯	◯
C	◯	◯	◯	◯	◯	◯
D	◯	◯	◯	◯	◯	◯
G	◯	◯	◯	◯	◯	◯

(v) What path is returned by greedy graph search?

◯ $S \to A \to B \to C \to D \to G$ ◯ $S \to A \to C \to G$ ◯ $S \to A \to C \to D \to G$
◯ $S \to A \to C \to D \to G$ ◯ $S \to A \to C \to D \to G$ ◯ None of the above

(b) Consider a complete graph, K_n, the undirected graph with n vertices where all n vertices are connected (there is an edge between every pair of vertices), resulting in $\binom{n}{2}$ edges. Please select the maximum possible depth of the resulting tree when the following **graph** search algorithms are run(assume any possible start and goal vertices).

	1	$\lceil \frac{n}{2} \rceil$	$n-1$	$\binom{n}{2}$	None of the above
BFS	◯	◯	◯	◯	◯
DFS	◯	◯	◯	◯	◯

(c) Given two admissible heuristics h_A and h_B.

(i) Which of the following are guaranteed to also be admissible heuristics?

☐ $h_A + h_B$ ☐ $\frac{1}{2}(h_A)$ ☐ $\frac{1}{2}(h_B)$ ☐ $\frac{1}{2}(h_A + h_B)$ ☐ $h_A * h_B$ ☐ $max(h_A, h_B)$ ☐ $min(h_A, h_B)$

(ii) Consider performing A* **tree** search. Which is generally best to use if we want to expand the fewest number of nodes?

◯ $h_A + h_B$ ◯ $\frac{1}{2}(h_A)$ ◯ $\frac{1}{2}(h_B)$ ◯ $\frac{1}{2}(h_A + h_B)$ ◯ $h_A * h_B$ ◯ $max(h_A, h_B)$ ◯ $min(h_A, h_B)$

(d) Consider performing tree search for some search graph. Let $depth(n)$ be the depth of search node n and $cost(n)$ be the total cost from the start state to node n. Let G_d be a goal node with minimum depth, and G_c be a goal node with minimum total cost.

(i) For iterative deepening (where we repeatedly run DFS and increase the maximum depth allowed by 1), mark all conditions that are guaranteed to be true for every node n that could be expanded during the search, or mark "None of the above" if none of the conditions are guaranteed.

☐ $cost(n) \leq cost(G_c)$
☐ $cost(n) \leq cost(G_d)$
☐ $depth(n) \leq depth(G_c)$
☐ $depth(n) \leq depth(G_d)$
◯ None of the above

(ii) What is necessarily true regarding iterative deepening on any search tree?

☐ Complete as opposed to DFS tree search
☐ Strictly faster than DFS tree search
☐ Strictly faster than BFS tree search
☐ More memory efficient than BFS tree search
☐ A type of stochastic local search
◯ None of the above