

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. [Optional] 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:

