

Mosab Mohamed – B20-04 – AI Assignment #1 Report

Algorithms Description:

- **Both Algorithms:**

Based on the assignment description, the optimal solution has to be one of the following:

Initial -> Book -> Exit

Initial -> Book -> Cloak -> Exit

Initial -> Cloak -> Book -> Exit

And since Harry has to get the Book before going to the Exit, the problem gets split into 2 problems.

1 - Get the book in the minimum number of steps

2 - Go to the Exit in the minimum number of steps

Therefore, I modified the algorithms so they could receive an Initial position and the Target they should be looking for.

And simulated the 3 cases step by step by calling the method of the algorithms for each step with a new initial position and a new target.

- **Backtrack:**

- The idea is to use a recursive function that explores the map, by trying all possible valid paths while counting the number of steps along the way.
- Since the number of recursive calls will be huge (even for 9x9 lattice, the unguided search is expensive and better algorithms exist for shortest path problems). I implemented an optimization to make the backtracking runs a little bit faster because in some cases the code would need to run for more than 30 minutes.
- The optimization implemented is to keep a memory 2d array that holds the shortest path we have to reach each cell, and if we go into that cell with a step count more than the one it already has, then there is a shorter way we found before so we return.

- **A*:**

- The idea is to use a heuristic function that guides us to our destination without us taking too many steps that would not benefit us.
- I used the euclidean distance as my heuristic function where I calculate the distance to the target from my neighbouring cells and take the one with the smallest number. We achieve that by using a priority queue to keep all the possible paths stored.
- The priority queue sorts our possible next steps based on the heuristic+the amount of steps it took us to get to that cell. And based on that we can choose the optimal step to take.

PEAS:

- **Performance measure:** the number of steps needed to get the book and reach the exit, and whether it is possible or not.
- **Environment:** 9x9 square lattice, representing physical spots.
- **Actuators:** Harry can move(legs) vertically, horizontally, and diagonally.
- **Sensors:** Harry can perceive(eyes) objects around him, from different distances based on the scenario.

Statistical analysis:

Note : the perception scenario for Harry did not introduce any output difference; Harry being able to see the inspectors from a large distance may affect his decision of going in a certain direction but will never affect the final shortest path.

Running 10 randomly generated maps and collecting the running time results(in seconds) :

#	Backtrack	A*
1	0.0000568	0.000888400013
2	0.001938899979	0.002767399885
3	0.004453099798	0.003697900102
4	0.002927399939	0.002791000064
5	0.005204500165	0.004356000107
6	0.003013300011	0.002739100019
7	0.000052899999	0.000824399991
8	0.00550869992	0.00336880004
9	0.005792899989	0.003950300161
10	0.0000569	0.000920500024
Σ	0.0290053998	0.02630380041

- **Backtrack:**

Mean Time = 0.00290053998 seconds

Median Time = 0.002970349975 seconds

Standard deviation = 0.0021933377729256

Generally: the process is almost always CPU intensive, due to the large number of recursive calls.

- **A*:**

Mean Time = 0.0026303800406 seconds

Median Time = 0.0027791999745 seconds

Standard deviation = 0.0012522363277538

Generally: the process is usually not CPU intensive, since the algorithm is efficient and the map is of dimensions 9x9

Graphical representation of custom maps:

- Easy Map:



Statistical Analysis:

Backtrack :

Outcome: Win

Number of steps: 8

Time taken: 0.001842899946 seconds

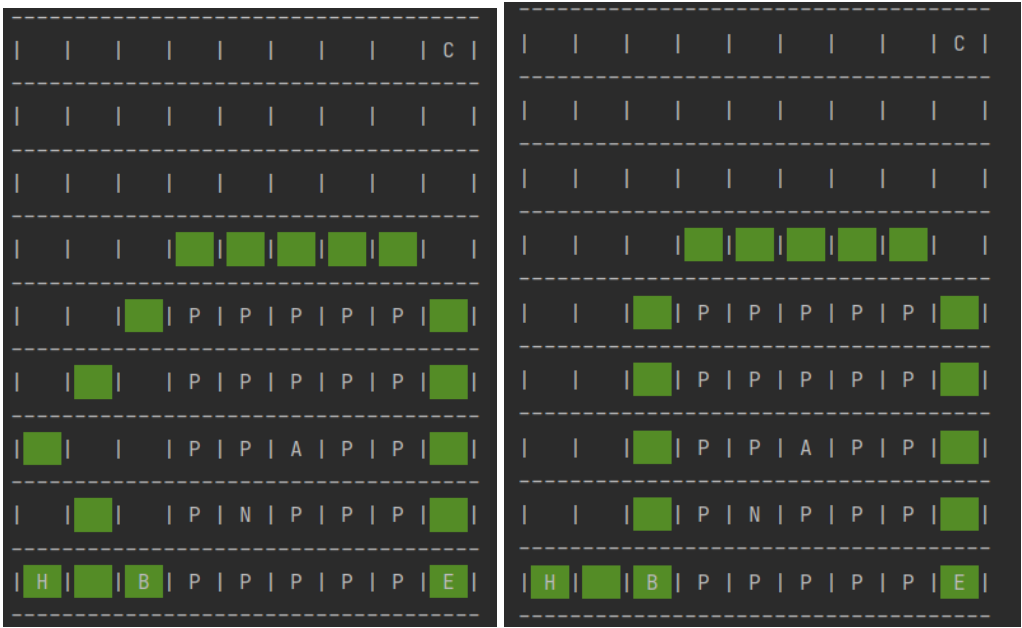
A* :

Outcome: Win

Number of steps: 8

Time taken: 0.002051100161 seconds

● **Medium Map:**



Statistical Analysis:

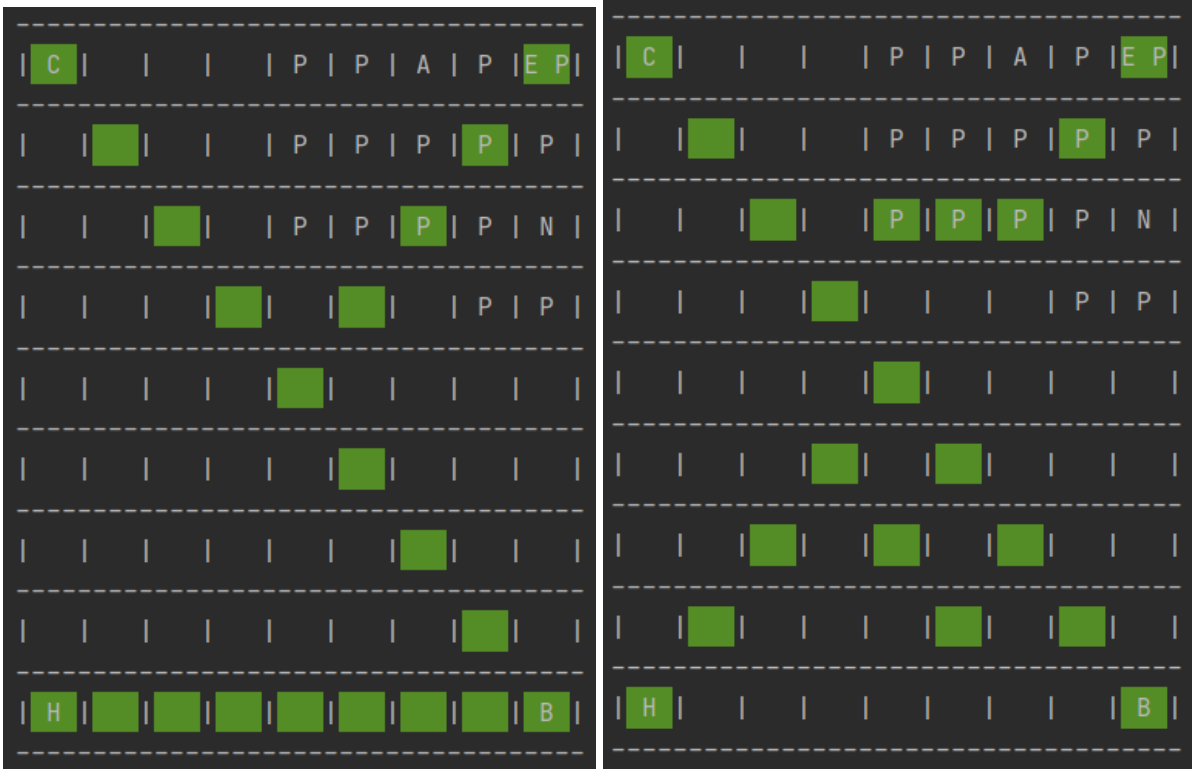
Backtrack(left) :

Outcome: Win
Number of steps: 16
Time taken: 0.003602999961 seconds

A*(right) :

Outcome: Win
Number of steps: 16
Time taken: 0.001745099995 seconds

● **Hard Map:**



Statistical Analysis:

Backtrack(left) :

Outcome: Win

Number of steps: 24

Time taken: 0.007085800171 seconds

A*(right) :

Outcome: Win

Number of steps: 24

Time taken: 0.004980800208 seconds

- Impossible Map:

			C						
	P		P		P				
	P		P		P			B	
	A		P		P				
	P		P		P		P		
	P		P		N		P		
	H		P		P		P		E

Statistical Analysis:

Backtrack:

Outcome: Lose

Time taken: 0.002186000114 seconds

A* :

Outcome: Lose

Time taken: 0.003323799931 seconds