Assignment 1 - Report

Dmitriy Okoneshnikov, B22-CS-01
d.okoneshnikov@innopolis.university
https://codeforces.com/profile/magicwinnie

November 6, 2023

Contents

1 Algorithm flow

The general idea for both of my algorithms is following:

1. Explore the map without picking up the Shield
2. Explore the map going through the cell with the Shield
3. Compare the two distances and choose the shortest

1.1 Backtracking search

Submission: https://codeforces.com/group/ux8yi0gQhD/contest/479327/submission/231522968

For backtracking I have used a modified version of Depth-First Search (described on lecture 5). Standard DFS was changed to allow the search of all paths rather than finding the first path to the goal. Some optimizations related to cutting off paths with worse distances to a cell were used. Also neighbours of a cell are visited in such order that those that are closer to the goal are visited first. All the information about the characters is stored in a map.

I start off by finding the shortest path (if any) from the start to the goal without picking up the Shield as it would be with us until the end of the test. If the Shield was found and it is accessible, then I move to the cell with it and run the algorithm once again to find the shortest path from the Shield to the goal. Finally, I compare the paths and find the shortest one.

The information about Thanos' perception is used to build perception zones of enemies, and, therefore, decrease number of cells to check.

1.2 A* search

Submission: https://codeforces.com/group/ux8yi0gQhD/contest/479327/submission/231523695

For A* I have used the algorithm described in Wikipedia (https://en.wikipedia.org/wiki/A*_search_algorithm#Pseudocode). The only change in the algorithm was to prevent teleportation: for each step I move back the interactor to the start and move from there to the new cell. All the information about the characters is stored in a map.

I start of by running A* to the Stone. If the Shield was found and can be reached then I run A* to the Shield and from it to the goal. Finally, I find the shortest path among the two.

The information about Thanos' perception is used to build perception zones of enemies, and, therefore, decrease number of cells to check.

2 Statistical comparison

For generating maps and running the solutions on them, a generator and an interactor scripts were written by me and tested by Matvey Korinenko. Note that these scripts were shared with other students. They can be found in the following Github Gist: https://gist.github.com/MagicWinnie/d8899001c41c851bdc1ae9a10def8fb4

The scripts were very helpful while debugging and counting the statistics. The generator creates n random maps following the rules stated in the problem. The interactor can be used with a solution on any language as it gets a command to run the solution with. Also it receives the directory with generated tests and the variant number. The interactor creates a CSV file with tests' file names, received answers from the solution, and the execution times. The CSV file then can be used to count the statistics manually or through other programs.

		Backtracking		A*	
		Variant 1	Variant 2	Variant 1	Variant 2
Time	Mean, s	0.096	0.099	0.078	0.086
	Mode, s	0.068	0.062	0.065	0.070
	Median, s	0.111	0.098	0.079	0.073
	σ, s	0.047	0.055	0.020	0.027
Wins	Count	914			
	%	91.4			
	Mean	914			
	Mode	914			
	Median	914			
	σ	0			
Losses	Count	86			
	%	8.6			
	Mean	86			
	Mode	86			
	Median	86			
	σ	0			

Figure 1: Comparison of statistics for all algorithms

The results in table 1 were received by running the interactor on the same 1000 tests for all combinations of algorithms and perception variants.

All of them gave the same answers for the tests. Therefore, number of wins, number of losses, and their percentages are equal for all columns. For this reason, mean, mode, median for count of wins/losses are equal to count of wins/losses. And standard deviations for count of wins/losses are equal to 0.

For counting modes the execution times were rounded to 3 digits after the decimal point.

2.1 Backtracking (variant 1) compared to A* (variant 1)

It is obvious from table 1 that A* outperforms Backtracking. In mean time A* is ≈ 1.2 times faster, but when we look at values that are not that affected by outliers, then actually A* is faster for ≈ 1.4 times. Also it is interesting to look at the standard deviations: it is ≈ 2 times larger for Backtracking, than for A*. Which means that A* works for different tests with closer times than Backtracking. Values of wins and losses were discussed above.

2.2 Backtracking (variant 2) compared to A* (variant 2)

It, basically, repeats the information given in subsection 2.1.

2.3 Backtracking (variant 1) compared to Backtracking (variant 2)

Mean time and standard deviation of time for variant 1 is smaller than in variant 2, but other values are larger. It means that for variant 2 there were some tests that executed with a more larger time, which led to increase in σ and mean time. But generally, variant 2 executed a little bit faster.

2.4 A* (variant 1) compared to A* (variant 2)

In A*, variant 2 did not give any time win as every value excpet median of time increased. This could mean that there is some problem in the implementation, but this should be investigated.

3 PEAS description

3.1 Performance Measure

The performance measure for Thanos would be reaching the Infinity Stone in the fastest way possible.

3.2 Environment

The environment for Thanos would be our map.

3.2.1 Properties of environment

Partially Observable. Thanos' vision is limited to his perception zone.

Multiple Agent. The agents are Thanos and Avengers. They all have sensors.

Deterministic. The next state of the map can be determined given the previous state and the action applying.

Sequential. Current decisions in our map can have consequences on future actions: picking up the Shield removes the perception zones of Hulk and Thor.

Static. When Thanos is staying still and is thinking about his actions, the environment does not change.

Discrete. We are operating with snapshots of the map and do not see, for example, Thanos smoothly moving from one cell to another.

Known. All the rules of the environment were given in the problem statement.

3.3 Actuators

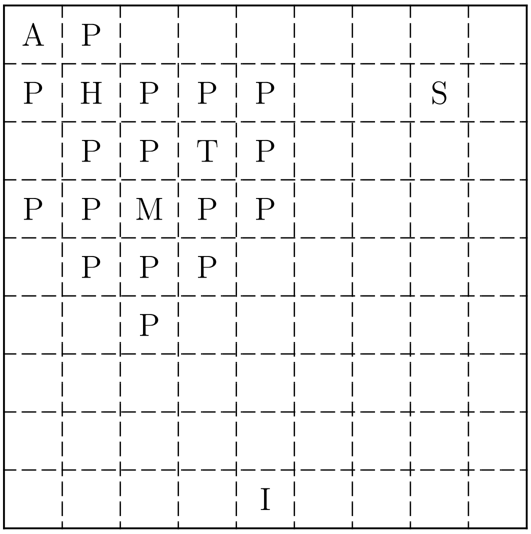
Thanos can move into adjacent cells, pick up the Shield and the Infinity Stone.

3.4 Sensors

Thanos has two variants of his vision (perception zone) that allow him to perceive characters (Avengers, Shield, Infinity Stone).

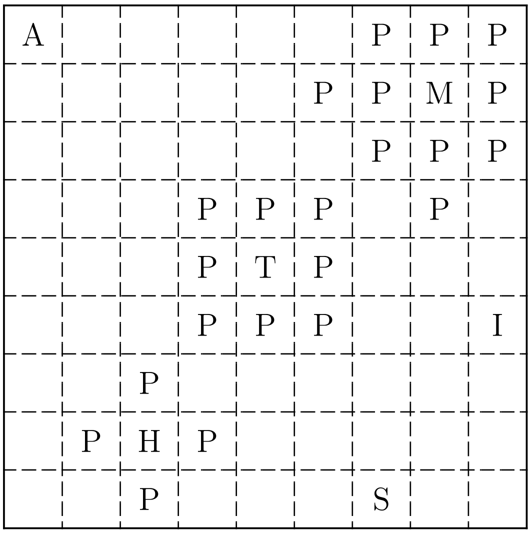
4 Impossible maps

These are some examples of impossible maps found by my solution (from 1000 of total maps generated):



A	P							
P	H	P	P	P			S	
	P	P	T	P				
P	P	M	P	P				
	P	P	P					
		P						
				I				

Figure 2: Cannot exit start cell



A						P	P	P
					P	P	M	P
						P	P	P
			P	P	P		P	
			P	T	P			
			P	P	P			I
		P						
	P	H	P					
		P				S		

Figure 3: Shield and Infinity Stone are unreachable from start

A				P	T	P		
				P	P	P		
		P						S
	P	P	P			P		
P	P	M	P	P	P	H	P	
	P	P	P			P		
I		P						

Figure 4: Shield is reachable, but Infinity Stone is surrounded by Captain Marvel

5 Interesting outcomes or maps

No interesting outcomes or maps were observed.