



Done by: Erokhin Evgenii
Group: DSAI-03
Email: e.erokhin@innopolis.university

A* algorithm:

A* algorithm search for a short path. Firstly we have stone as an input(coordinates); We start from point (0;0) say that we have made a move; then count distance from (0;0), distance to stone and its sum for near cells which have no obstacles. Then make a move, always choose a cell according to a sum(smaller); Then go to the cell with the smallest sum. Repeat the algorithm while we have not visited cells and choose the smallest path. Count distance(it already counted in sum)

Backtracking algorithm:

Backtracking algorithm search for all possible path from Tanos to Stone and chose the shortest When I use a backtracking algorithm. Firstly I search for the shortest path from Tanos to Stone. Then I search for the shortest path from Tanos to Shield and then path from shield to stone, sum them and compare it to the path without it. Take the smallest. Print result

A-star algorithm: 1000 tests generated for variant = 1 and variant = 2

A*	1st var	2nd var
number of wins	904	904
number of loss	96	96
Percentage of wins	90,4%	90,4%
Percentage of loss	9,6%	9,6%
Mean	0,06905s	0,07153s
Median	0,06581s	0,06791s
Mode	0,06136s	0,05072s
Standard deviation	0,02126s	0,02597s

Backtracking algorithm: 1000 tests generated for variant = 1 and variant = 2

Backtracking	1st var	2nd var
number of wins	901	901
number of loss	99	99
Percentage of wins	90,1%	90,1%
Percentage of loss	9,9%	9,9%
Mean	2,58986s	2,37574s
Median	0,09738s	0,07342s
Mode	0,03341s	0,02207s
Standard deviation	7,96347s	7,26439s

PEAS description:

Performance Measure - find shortest path from Tanos to Stone

Environment - grid of size 9x9 with obstacles

Actuators: Tanos

Sensors: perception zone

Partially Observable

Deterministic

Sequential
 Dynamic
 Discrete
 Multiple agent
 Unknown

Compare:

- 1) A* first var with Backtracking first var
 Firstly we can see than A*(1) has more wins than Backtracking(1) also A*(1) works quite faster
- 2) A* second var with Backtracking first var
 Firstly we can see than A*(2) has more wins than Backtracking(1) also A*(2) works quite faster, but slower than A*(1)
- 3) A* first var with Backtracking second var
 Firstly we can see than A*(1) has more wins than Backtracking(2) also A*(1) works quite faster but not so slow as B(1)
- 4) A* second var with Backtracking second var
 Firstly we can see than A*(2) has more wins than Backtracking(2) A*(2) slower than A*(1)
 Backtracking(2) faster than Backtracking(1). And $\text{Backtracking}(2) < A^*(2)$

Example of unsolvable maps:

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

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

Interesting case when we put Tanos on (0;0); but the shield is already so we should check it somehow

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

It is solvable map