

# Introduction to Artificial Intelligence

## Assignment 1

Anton Nekhaev BS21-07,  
a.nekhaev@innopolis.university

Fall 2022

### 1 Assumptions

1. Agent can not go out of bounds of field.
2. Cost of horizontal, vertical and diagonal moving is 1 point.
3. When we randomly generate the map an agent spawns in  $(0, 0)$  coordinate.

### 2 PEAS of Jack Sparrow

Performance measure - achieving the Dead Man's Chest with a shortest existed path.

Environment - matrix  $9 \times 9$

Actuators - changing position on map, visiting Tortuga, killing Kraken

Sensors - spyglass (depends on scenario), compass

### 3 Properties of the environment

Partially Observable - Jack Sparrow can see only in his perception zone.

Single Agent - only agent can move on the map, other entities are static.

Sequential - the current decision has consequences on future actions, so, if Jack has killed the Kraken he can move in Kraken's coordinates without damage.

Dynamic - agent can change the environment by killing Kraken.

Discrete - Jack Sparrow makes discrete steps from cell to cell.

Known - our agent has a compass, so he knows where Dead Man's Chest and Tortuga Island are located.

### 4 Search algorithms

#### 4.1 Backtracking search

Backtracking search tries all paths and finds for shortest one. For the implementation I use recursive approach. Also I have a two dimensional array - *map* with capacity  $9 \times 9$ , cell with index  $(x, y)$  corresponds to the minimal length of path that needed to get to this coordinate. So, thanks to presence of map array we can make a optimization of algorithm: if on one step current path to  $(x, y)$  coordinate is greater than value of *map* $[x][y]$  then we can exit from this recursion leaf because every other path in this recursion branch will be greater than path that we have obtain on previous steps. Recursion ends when can not make move to another cell or speaking in another words: we can not make solution for adjacent cells better than already exists.

## 4.2 A\*

A\* wants to reach the target cell (if possible) from the starting cell as quickly as possible, comparing to Backtracking search A\* does not tries all paths. To do that it, he goes to cell with the minimum value of  $f$ ,  $f = g + h$ , where  $g$  - amount of steps from initial start position and  $h$  - heuristic value, the minimum number of steps you need to take to be on the Dead Men's Chest Island if you go regardless of opponents. So, for the implementation I use a priority queue, node of which can be compared according to  $f$  and if  $f$  values are equal, nodes will compare according to  $h$  values. So, on each step we will pick the node with the least cost, try to go to adjacent cells if cost to them that we obtained on previous steps is greater then current cost. For the first time when we reach the Dead Men's Chest Island this path will be the answer. Also for optimization I use the same approach as in Backtrack search (read above).

## 5 Solution of the problem

For solving the specific board I consider to ways of reaching the Dead's Men Chest:

1. Direct way to Dead's Men Chest Island.
2. Way to Dead's Men Chest Island through Tortuga island to ensure that Jack have ability to kill the Kraken on his way from Tortuga to Dead's Men Chest Island.

## 6 Maps

In this section I propose to consider some notable maps.

### Map legend:

J – Jack Sparrow

T – Tortuga

C – Dead's Men Chest

K – Kraken

R – Rock

DJ – Davy Jones

\* – Path

### 6.1 Interesting map

In this subsection there is an interesting map. We need to go to Dead's Men Chest Island through Toruga Island.

x/y	0	1	2	3	4	5	6	7	8
0	J / *								
1		*							
2			*						
3				*					
4		DJ		*					
5				*					
6			*				*		
7		*	*	*	*	*		K / *	
8	T						R		C / *

(a) Remarkable map A

## 6.2 Maps impossible to solve

In this subsection there are maps (without their variations) that can not be solved using both algorithms, because on each map path to Dead's Men Chest does not exist.

x/y	0	1	2	3	4	5	6	7	8
0	J								
1									
2									
3									
4									
5									
6									
7									
8	T								

(a) Impossible map A

x/y	0	1	2	3	4	5	6	7	8
0	J								
1									
2									
3									
4									
5									
6									
7									
8	T								

(c) Impossible map C

x/y	0	1	2	3	4	5	6	7	8
0	J								
1									
2									
3									
4									
5									
6									
7									
8									

(b) Impossible map B

x/y	0	1	2	3	4	5	6	7	8
0	J								
1									
2									
3									
4									
5									
6									
7									
8	T								

(d) Impossible map D

- (a) – Even if we kill the Kraken the path will be crossed by the stone.
- (b) – It is impossible to visit the Tortuga to kill the Kraken afterwards.
- (c) – Stone prevents the kraken from being killed.
- (d) – Instance death for Jack Sparrow.

## 7 Statistical Analysis

The analysis is based on 1000 runs of program, each program executes algorithms in 2 scenarios, that generate random map, each of which satisfies the task conditions. Let's compare!

### Backtracking compared to A\*

	Backtracking	A*	Backtracking	A*
Scenario number	1		2	
Mean of time for execution (ms)	3.647	0.694	2.627	0.306
Mode of time for execution (ms)	3.62	0.63	2.64	0.29
Median of time for execution (ms)	3.629	0.677	2.666	0.296
Standard deviation of time for execution (ms)	3.662	0.712	2.674	0.329
Number of wins	942			
Number of loses	58			
Percentage of wins	94.2%			
Percentage of loses	5.8%			

As we see from the table A\* solves this problem in less time than Backtracking in both scenarios, but algorithms show themselves the same when finding a path (if one algorithm finds a path, the second will find it too).

Also notable, if you compare the speed of the algorithm under different scenarios, you can see that the second scenario works faster than the first.

## 8 Source code

Java source code, together with files and python script for statistics are available on my Github repository, the link (just click on it).