INTRODUCTION TO ARTIFICIAL INTELLIGENCE.

ASSIGNMENT 1.

Output.txt: contains the time to solving the algorithm and which algorithm is quicker

Console: contains map and how much step algorithm takes

PEAS

Red Riding Hood:

Agent type:

Red Riding Hood (Goal-based agent) the girl may change her goals, depending on the he current state.

Performance measure:

To reach granny, with full basket (6 berries)

Environment:

Forest (2-Dimensional 9x9 lattice)

Actuators:

Move up, down, left, right

Avoid bear, wolf

Take risk

Go to granny

Find woodcutter

Sensors:

Hearing the howl of the wolf

Hearing the smell of the bear

Knowing of losing the berries

Finding the woodcutter’s work/house empty

Understanding of reaching granny

Wolf:

Agent type:

Wolf (Simple-reflex agent) wolf only responses to its sensors

Performance measure:

To kill Red Riding Hood

Environment:

Forest (2-Dimensional 9x9 lattice)

Actuators:

Kill Red Riding hood

Sensors:

Sensing the Red Riding Hood

Bear:

Agent type:

Bear (Simple-reflex agent) bear only responses to its sensors

Performance measure:

Eat berries

Environment:

Forest (2-Dimensional 9x9 lattice)

Actuators:

Steal, than eat two berries from Red Riding Hood basket

Sensors:

Sensing the Red Riding Hood

Granny:

Agent type:

Granny (Simple-reflex agent) granny only responses to her sensors

Performance measure:

Meet Red Riding Hood

Environment:

Forest (2-Dimensional 9x9 lattice)

Actuators:

Take berries and end the game

Sensors:

Seeing Red Riding Hood (when she is in the same cell)

Woodcutter:

Agent type:

Woodcutter (Simple-reflex agent) woodcutter only response to his agents

Performance measure:

To save Red Riding Hood (gave her all the berries back)

Environment:

Forest (2-Dimensional 9x9 lattice)

Actuators:

Working

Sitting at home

Gave berries to Red Riding Hood

Sensors:

Seeing Red Riding Hood (when she is in the same cell)

PROPERTIES OF THE ENVIRONMENT (FOREST)

- Partially Observable:

  All agents does not know all the map, only the limited information, which gives them their sensors
- Multiple Agent:

  There are multiple agents in the environment. But only one moving agent – Red Riding Hood. However, each agent have some action and demand cooperation with the Red Riding Hood.
- Stochastic:

  All the steps of the Red Riding Hood are not predictable, and are based upon some level of randomness.
- Sequential:

  All the agents' actions have the consequences on future actions. If Red Riding Hood goes the wrong way, she may lose all her berries or even die. Bear and Wolf may attack the Red Riding Hood. If Bear eats her berries, Red Riding Hood must go to the woodcutter, and choose, in which of two places to go and so on.
- Static:

  The environment does not change, while the agents think about their actions. Unmoving agents does not change their position, while Red Riding Hood thinks, where to go. And Red Riding Hood cannot go out of wolf or bear detection range's while they are thinking if she is there or not.
- Discrete:

  There is finite number of distinct states of the environment, which does not evolve or change over time, based on some physical laws.
- Known:

  Whole agents know the rules of the environment, even though may not know the while environment itself. So, Red Riding Hood knows that she can lose her berries, while stepping on the Bear detection range, or can be killed, if she tries to go through the Wolf and so on.

VERY DIFFICULT AND UNSOLVABLE MAPS

There are some different mappings of the agent, in which map becomes very hard or even unable to solve.

1. The Wolf and Bear blocked the Red Riding Hood from woodcutter and Granny (Very difficult solvable map)

R								
			B					
w								
				G				
						C		
		H						

r							g	c
					b			
								w
			f					

And other different variations of the same situations

2. The Wolf blocked Red Riding Hood form other agents (Unsolvable)

r								
	w							
					g			
		b						h
				c				

And other different variations of the same situations

3. The Wolf blocked Granny from other agents (Unsolvable)

r								
						c		
			b					
	h							
							w	
								g

r								
				c				
			b					
							f	
	w							
g								

And other different variations of the same situations

4. The Wolf and Bear blocked Granny from other agents (Unsolvable)

r								
	c		b					
w								
					f			
				g				

r								
				h				
		c						
								w
						b		
								g

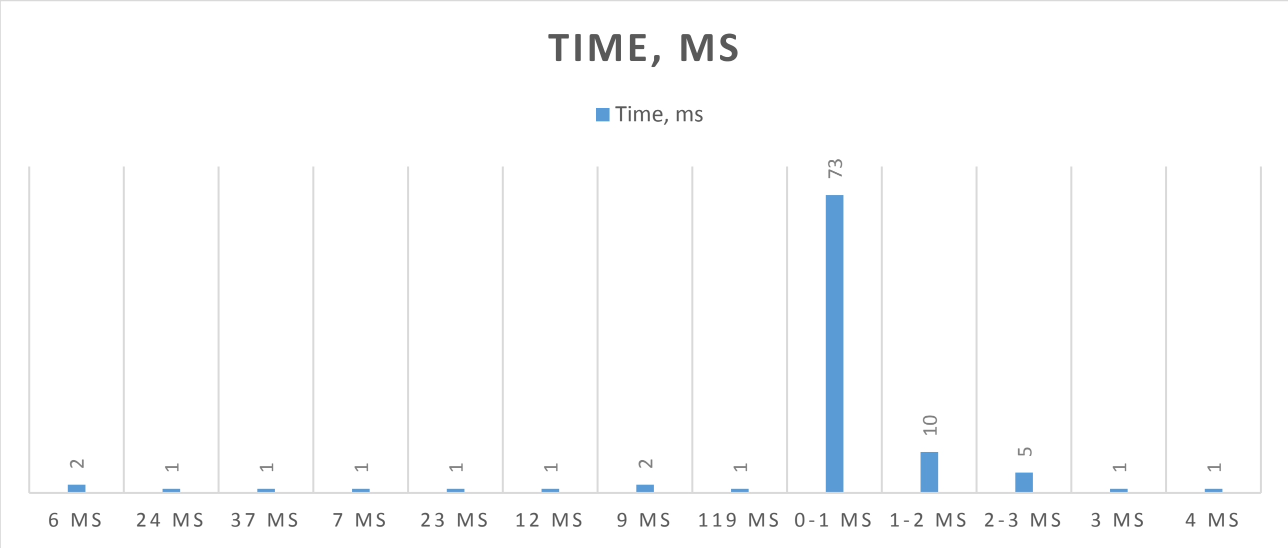
And other different variations of the same situations

BACKTRACK ALGORITHM

One of the algorithm, that was used to solve this problem, is Backtrack algorithm. Red Riding Hood goes up or right or down or left, until meets the dead-end or reaches her goal. If there's a dead-end, Red Riding Hood goes to the last step and goes other way. The algorithm returns, whether she found the granny or not.

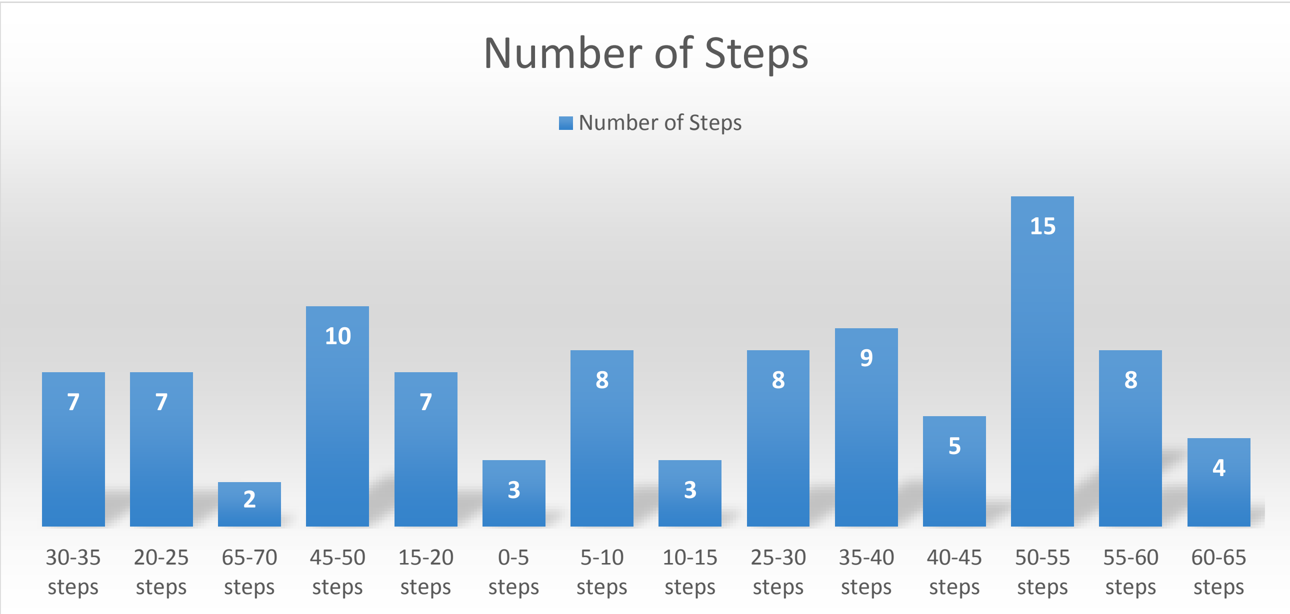
I modified the algorithm, so Red Riding Hood may change her goal, if she doesn't find the Granny, In this situation, Red Riding Hood goes to the last cell, in which she met the bear, and tries to go through him. If it was successful, she tries to find woodcutter's work, then his house, if woodcutter was not at work. If it was not successful, she lost the game.

2 from 100 maps the Red Riding Hood lost



Mean of solving the algorithm = 3,048 ms

Mode = 0-1 ms



Mean number of Steps = 36,03 steps

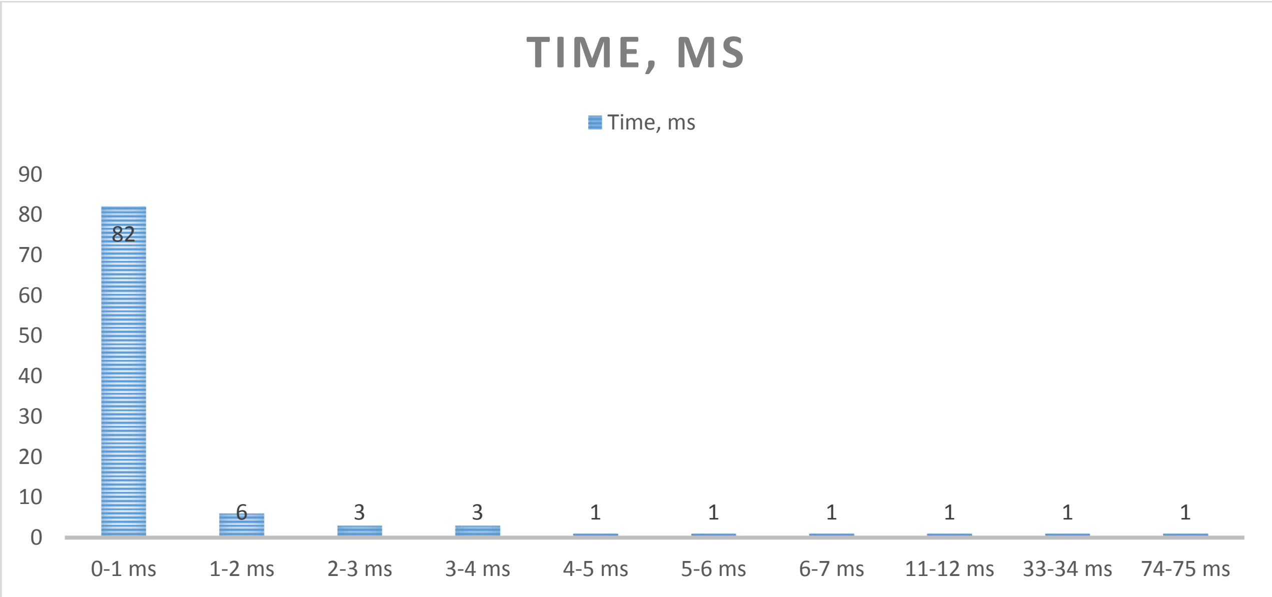
Mode = 50-55 steps

A* ALGORITHM

The second algorithm, that was used in this problem, is A* algorithm. Red Riding Hood now goes specifically to the Granny, counting the cost of each her move, in order to find the shortest path.

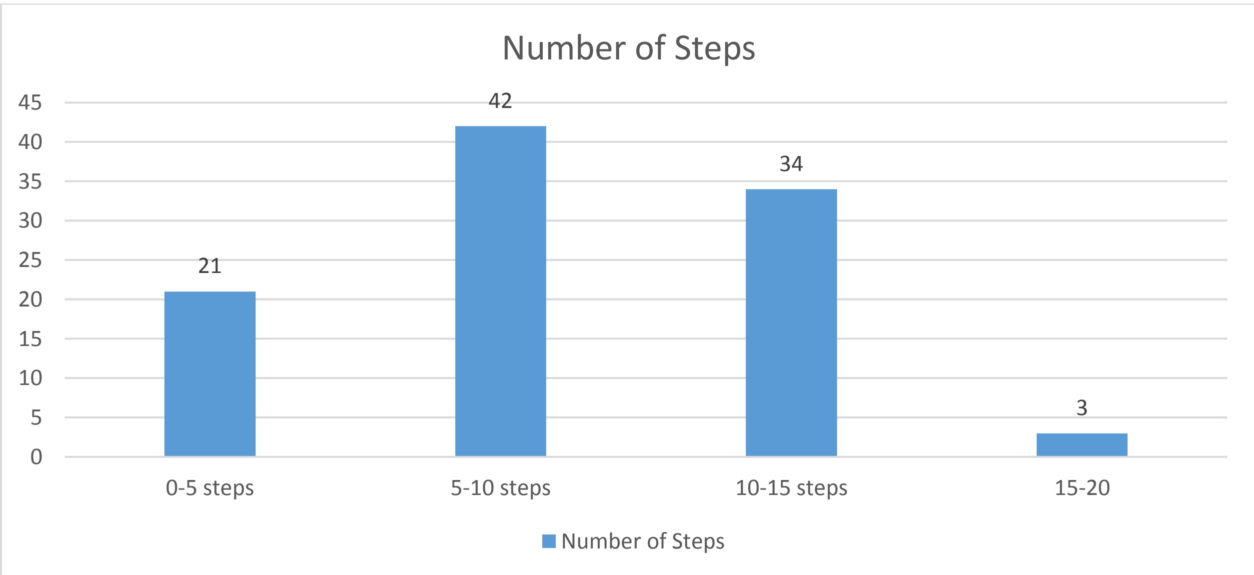
I modified the algorithm, so Red Riding Hood may change her goal, if she doesn't find the Granny, In this situation, Red Riding Hood goes to the last cell, in which she met the bear, and tries to go through him. If it was successful, she tries to find woodcutter's work, then his house, if woodcutter was not at work. If it was not successful, she lost the game.

2 from 100 maps the Red Riding Hood lost.



Mean = 3,1156 ms

Mode = 0-1 ms



Mean = 9,07 steps

Mode = 5-10 steps

COMPARISON

As we can see both algorithms on average time work the same time, although Backtrack works a little bit more faster.

From 100 tests, 39 times of it A* algorithm worked quicker than Backtrack.

The biggest difference = 118,7 ms.

The average difference = 6,24 ms

44 times out of 100 Backtrack algorithm worked quicker than A* algorithm

The biggest difference = 74,8 ms

The average difference = 3,3 ms

17 times out of 100 their time was equal.

A* algorithm searches the quickest way to the Granny, so its number of steps is obviously less than Backtrack algorithm's number of steps. There's no outcome out of 100 tests, where Backtrack algorithm had less steps than A* algorithm.

19 times out of 100 number of steps were equal.