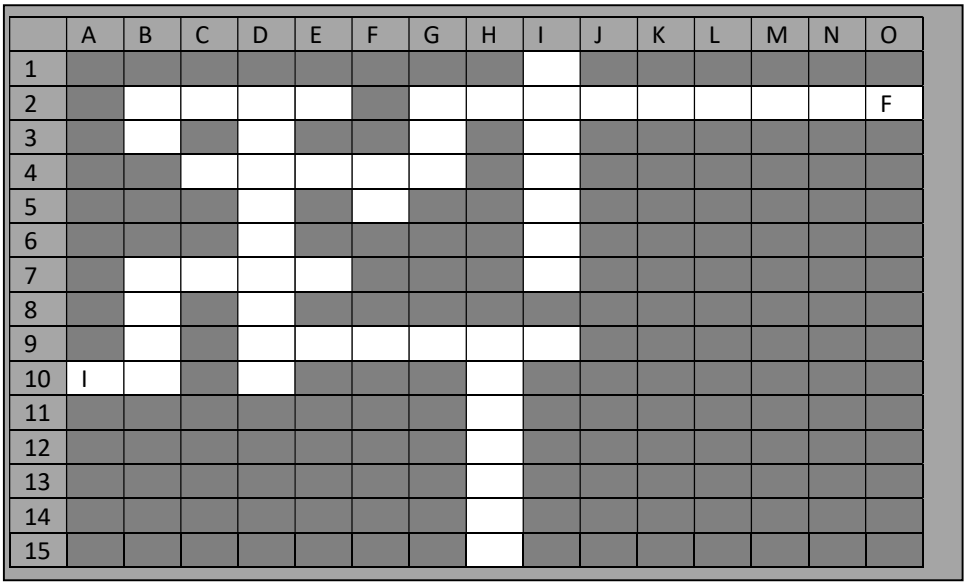
2ND LAB PRACTICE: SOLVING LABYRINTH USING BREADTH FIRST SEARCH, DEPTH FIRST SEARCH AND ITERATIVE DEPTH FIRST SEARCH

In this practice you would solve automatically labyrinths using Breadth-first search, Depth-first search and Iterative Depth-first search.

Add to the system previously developed the ability to solve automatically those kinds of labyrinths, first is necessary to specify the Initial point and the final point, for example in the following map:

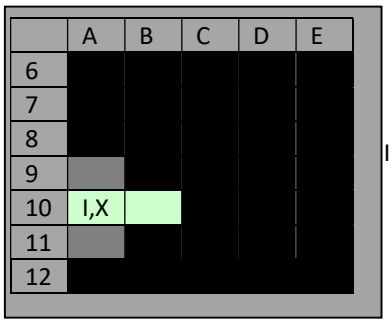


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2															F
3															
4															
5															
6															
7															
8															
9															
10	I														
11															
12															
13															
14															
15															

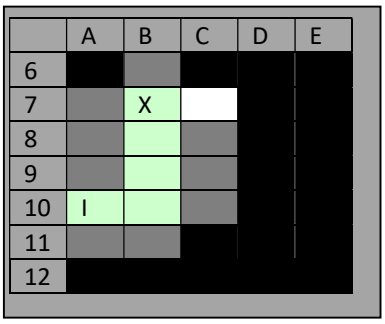
The coded used is:

Value	Means
I	Initial point
F	Final point

Also consider that a being X is collocated inside the labyrinth, is able to memorize the labyrinth the map of the labyrinth already visited and is able to see only one cell of distance (up, down, left and right). Initially the being inside doesn't know all the map and construct it travelling over it, and discovering only what he can see. For example, in the following labyrinth the visited cell are marked with green and the unknown map with black:



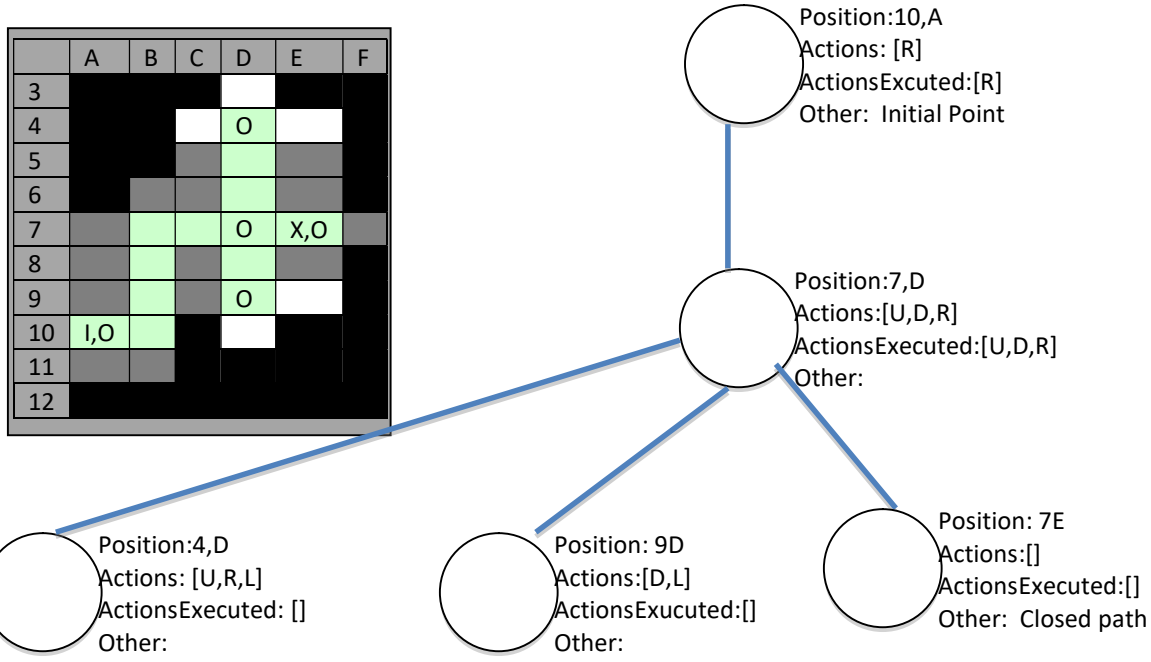
Initial



After 4 steps

Each time that decisions have to be made is marked in the labyrinth in some way and creates a node in the solution tree, in the following map is marked with O, the initial point is also considered as a node due that we need to save the position and also the end of a closed path are marked as nodes.

For example using Breadth-first search using the directional criteria (up, down, left, right) in a given time the discovered map and the tree generated would be as follows



Develop a system that:

1. Establish a directional priority for the being inside the labyrinth.
2. Establish the initial and final point of the labyrinth
3. Solve automatically the labyrinth using:
    a. Breadth-first search
    b. Depth-first search
4. Show the actual discovered map of the labyrinth
5. Show the actual tree generated
6. The system have to bring the option to show how is solved the labyrinth using one of the following options:
    a. Step by step (considering each cell as node in search tree)
    b. Decision by decision (only cells in wich a decision is necessary are nodes in search tree)