The map contains a starting point and an ending point. There are walls around the labyrinth and inside it. The map also contains portals, which teleports the player to a destination with a given probability. The game ends when the player arrives at the destination.

Task 1:
Implement an agent to find the minimum path using A*. For this task, do not use the portals.
Task 2:
Modify the previous algorithm to use the portals and their destinations probabilities. Compute the average number of actions needed after 1000 rounds.
Task 3:
The agent knows both the map and the positions of portals, but he does not know their destinations and neither the probabilities to arrive to those destinations. He has some time to explore the map and get the missing information: 100, 1000 and 10000 of exploring steps. Compute the average number of actions needed to arrive at destination for each exploration: 100, 1000, 10000.

To run the sources:

python Labyrinth_portals.py *map_name*

Readme:

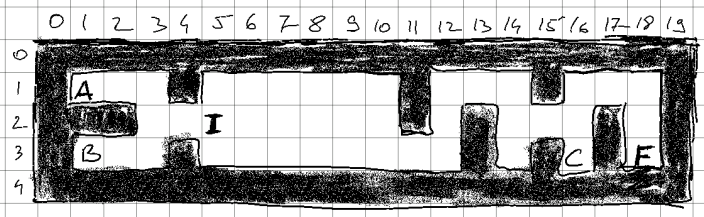
For the **first task**, I implemented A* algorithm.
For the **second task**, I modified the previous algorithm to include the portals and information they bring: destinations and probabilities to arrive to the respective destination. The new Heuristic becomes:

h = $dist(crt, portal) + p1 * dist(dest1, fin) + p2 * dist(dest2, fin) + ...$

H is applied to all th portals and the portal with th minimum value is chosen. The minimum is also compared with the direct distance to the end goal.
For the **third task**, I created a method which goes through the labyrinth and:

1. Chooses the closest portal
2. Applies A* to arrive to the portal
3. From the possible destinations, choose the closest one to another portal, or the one with fewest access times, to ensure an equal access to all portals visited.
4. Repeat steps 2-3 until the training steps are finished (100,1000,10000).
5. Apply the algorithm from the second task to find the average number of actions needed to arrive at the destination

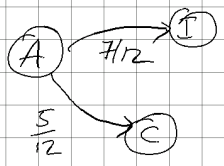
Next are some maps used for testing purposes:

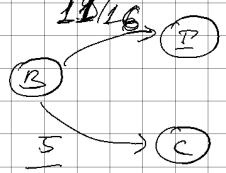


Stare inițială : I

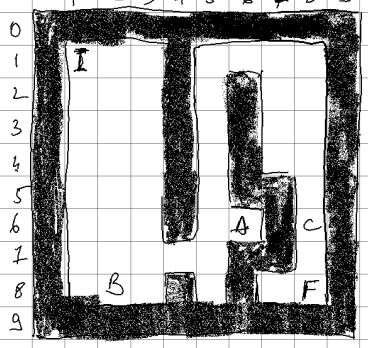
Stare finală : F

Porți : A, B





harta 3



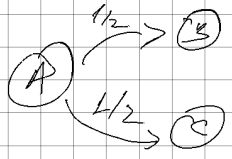
Stare inițială: I ; Stare finală: F

Porți : A

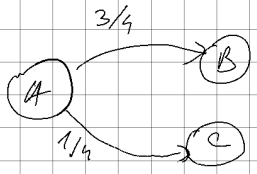
harta 2

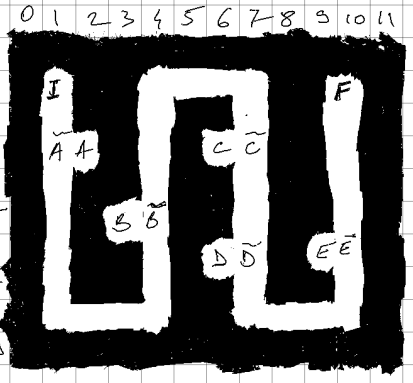
harta 1

Cazul 1



Cazul 2





[IP] = 37 (done)

	$\tilde{A}$	$\tilde{B}$	$\tilde{C}$	$\tilde{D}$	$\tilde{E}$
A	0	0.25	0.25	0.25	0.25
B	0.25	0	0.25	0.25	0.25
C	0.25	0.25	0	0.25	0.25
D	0.25	0.25	0.25	0	0.25
E	0.25	0.25	0.25	0.25	0

Stare inițială : I

Stare finală : F

Porți : A, B, C, D, E

harta 4