Python 3 Days Challenge (PODC)

An Amazing Problem to Solve

— EPITA —

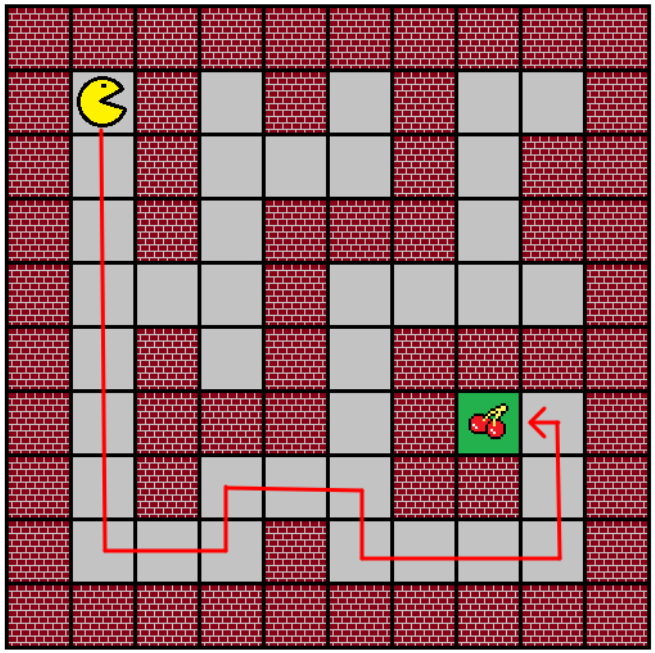
M. Angoustures & R. Dehak & R. Erra
& M. Jivalian & A. Letois

April 2020

Your main goal :

You have to solve a maze.

- You start from a fixed position in the maze.
- You need to find the fastest way to reach the fixed end point.



Example of a maze solved

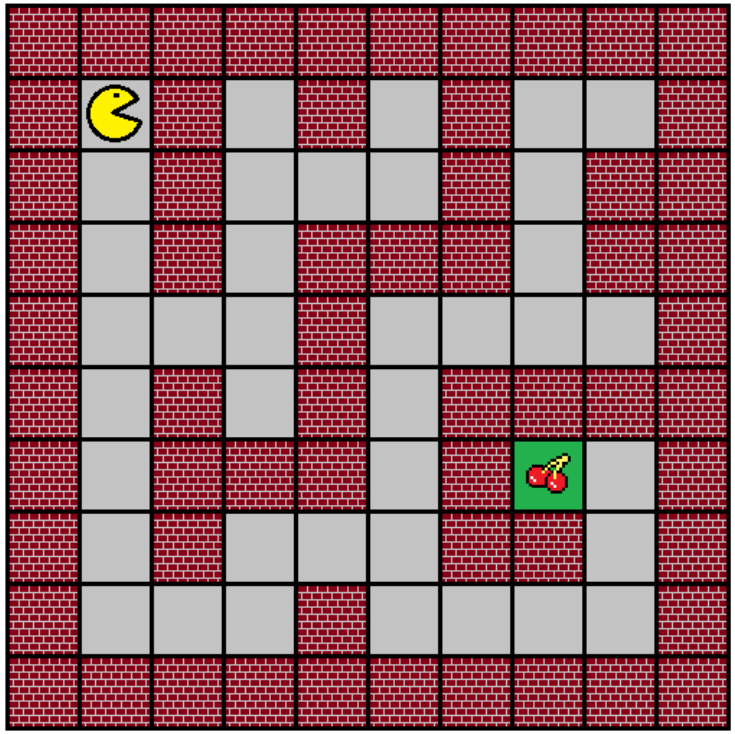
What do you have as input ?

You have a matrix which describe every element of the maze

```
1 1 1 1 1 1 1 1 1 1
1 s 1 0 1 0 1 0 0 1
1 0 1 0 0 0 1 0 1 1
1 0 1 0 1 1 1 0 1 1
1 0 0 0 1 0 0 0 0 1
1 0 1 0 1 0 1 1 1 1
1 0 1 1 1 0 1 e 0 1
1 0 1 0 1 0 1 1 0 1
1 0 0 0 1 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
```

Matrix format of the maze

=



Picture of the respective maze

Matrix description

Main element of the matrix has a rule in maze

Rules :

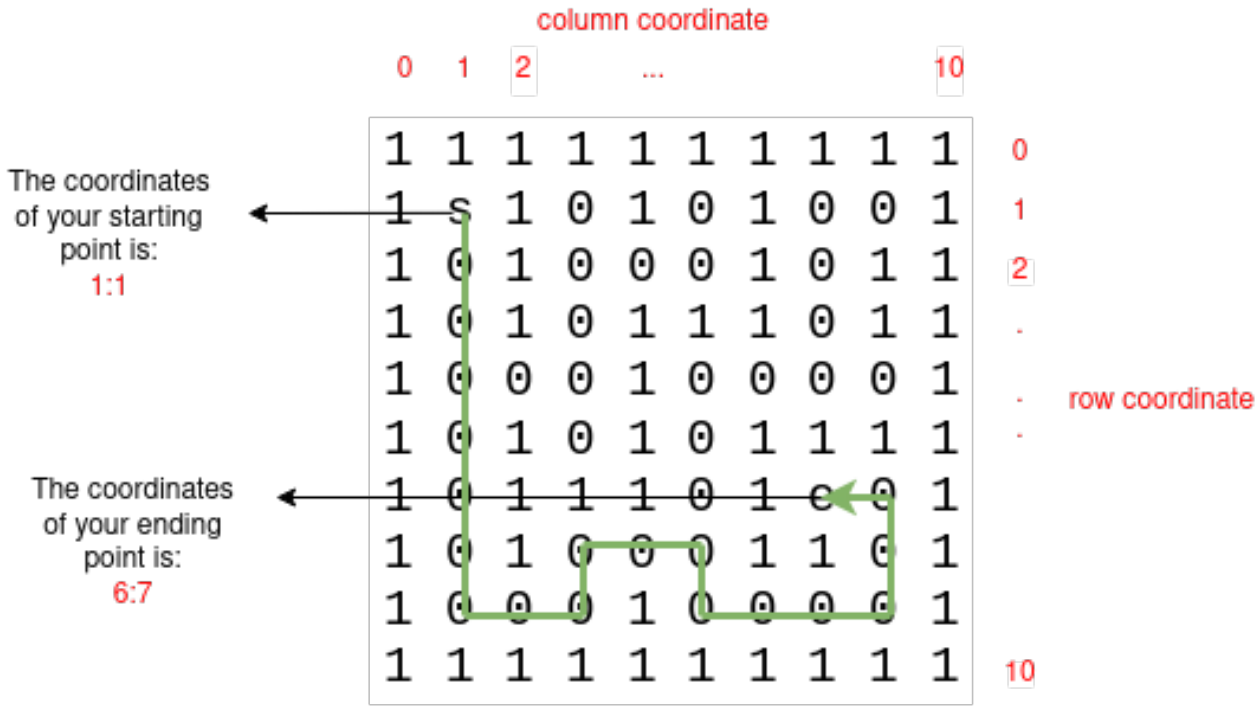
- **0 : path**
  you have to move this block
- **1 : wall**
  you cannot move to this block
- **s : start**
  your starting point
- **e : end**
  your ending point

```
1 1 1 1 1 1 1 1 1 1
1 s 1 0 1 0 1 0 0 1
1 0 1 0 0 0 1 0 1 1
1 0 1 0 1 1 1 0 1 1
1 0 0 0 1 0 0 0 0 1
1 0 1 0 1 0 1 1 1 1
1 0 1 1 1 0 1 e 0 1
1 0 1 0 1 0 1 1 0 1
1 0 0 0 1 0 0 0 0 1
1 1 1 1 1 1 1 1 1 1
```

Example

What do we expect as a result ?

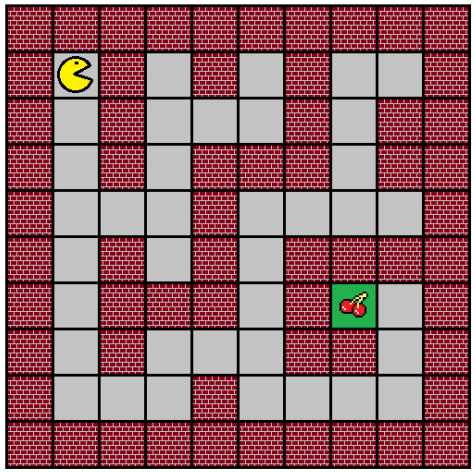
You need to provide a **list of tuples** which describe :
the matrix coordinate point of the path from start to the end

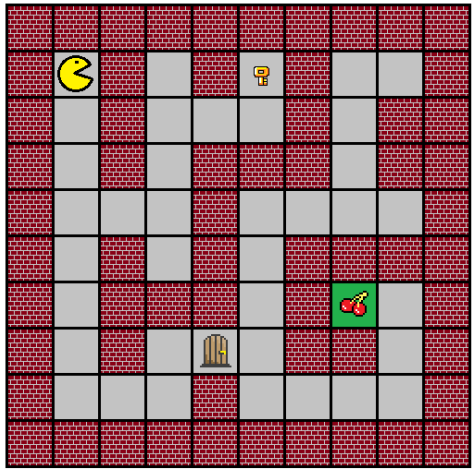


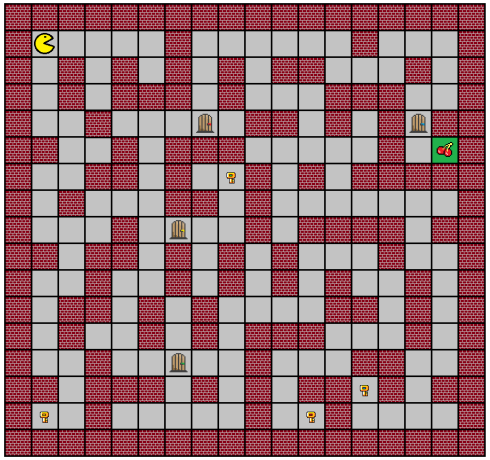
Your result:

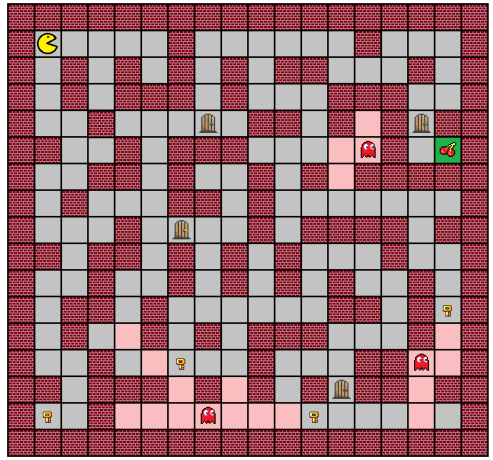
```
[(1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1),
(7, 1), (8, 1), (8, 2), (8, 3), (7, 3),
 (7, 4), (7, 5), (8, 5), (8, 6), (8, 7),
 (8, 8), (7, 8), (6, 8), (6, 7)]
```

4 types of maze to solve : from easiest to hardest









Different obstacles to overcome

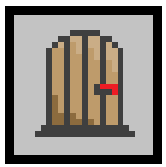
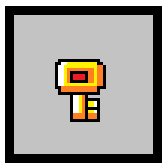
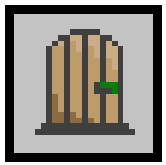
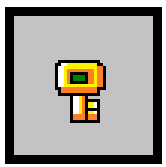
Image	Matrix notation	Description
	g	A red door prevent to move forward until you have the red key to open it.
	f	A red key which let you to open the red door
	c	A green door prevent to move forward until you have the green key to open it.
	d	A green key which let you to open the green door

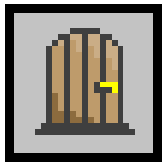
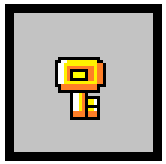
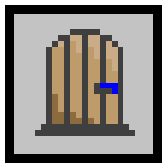
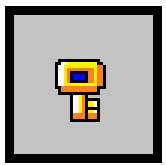
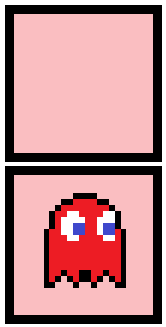
Image	Matrix notation	Description
	b	A yellow door prevent to to move forward until you have the yellow key to open it.
	a	A yellow key which let you to open the yellow door
	i	A blue door prevent to to move forward until you have the blue key to open it.
	h	A blue key which let you to open the blue door

Image	Matrix notation	Description
	2 or more	The ghost has a range of 2 cells or more (pink in the pictures examples) which kill you in all direction if you move on it. Avoid it!!!

Your final result

Program a solver to compute the fastest path from start to the end by :

- Finding the right color key for right color door
- Avoiding the ghost's line of sight.

If you find different paths, return the shortest. Don't forget : you always must return a **unique** path, i.e. a unique list of tuples.

Expected usage and result example

python solvemaze.py maze1.txt will give us :
[(1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1), (7, 1), (8, 1), (8, 2), (8, 3), (7, 3), (7, 4), (7, 5), (8, 5), (8, 6), (8, 7), (8, 8), (7, 8), (6, 8), (6, 7)]

Warning

- DO NOT HARD CODE THE PATH!!!
- We will test much bigger matrix matching completely new mazes.
- Your program will be able to find each path for each new matrix (input data).

Your secondary goal :

You have to program the game interface.

- You could use Pygame or Pygame zero.
- You need to animate the moving of the *Pacman* to the end.

We will provide you the images matching the element of the maze.

Where to sent your work : 1/2

Here are the instruction to submit

1. your work
2. your video.

Sent [a] to p3dcmsc@protonmail.com

1. Your python code
2. **and** your Jupyter Notebook if you have one.
3. and (mandatory) : add **all** your Family and First names both in your python code and in your Jupyter Notebook if you have one.
4. A unique email please !
5. **Don't forget to give your Team Number please.**

---

a. Don't use podcmsc@protonmail.com !

Where to sent your work 2/2

With your code, you have to present your results in the form of a mini video by respecting thoroughly the below rules :

1. The video should be 5 minutes long broken-down as follows :
2. 1st minute : team introduction and presentation of each part(s) done by each team member. We should see each team member and hear his/her name and the number of team.
3. No need to recall the problem, we all know it very well (gain time).
4. Describe your overall strategy. Explain your optimization methods. Express your overall results.
5. A short Interface demonstration.
6. Deadline : Monday, the 4th of Mat 04 May 2020 11h42 Paris time. Upload your video to Teams (see assignment Python Week # 2 team)
7. Please no video by email !

Actually : a suggestion . . .

For security reasons, don't forget to sent your code by EMAIL and to upload it (zip file could be nice) on Teams (see assignment Python Week # 2 team).

Get the best score you can !

. . . **Congratulations and** *bon courage* **to all of you for the last steps.**