Wild Pig Maze by Nob Yoshigahara

An algorithmic approach

Do Xuan Anh
Le Quang Dung
Hoang Anh Quan
Trinh Huy Vu

December 8, 2020

Table of content

Problem statements

Problem statements

Figure 1: Nob Yoshigahara, Puzzles 101: A PuzzleMasters Challenge

Algorithm

Overview

1. Creating a $(0, 1)$ - matrix from the image of the maze.
2. Changing matrix to a directed graph.
3. Find a path of the directed graph and draw it to the image.

Step 1: Creating a $(0,1)$ - matrix from the image of the maze

Consist of THREE main stages:

1) Using PIL package to save the pixels of the image as a $(0,1)$ - matrix.

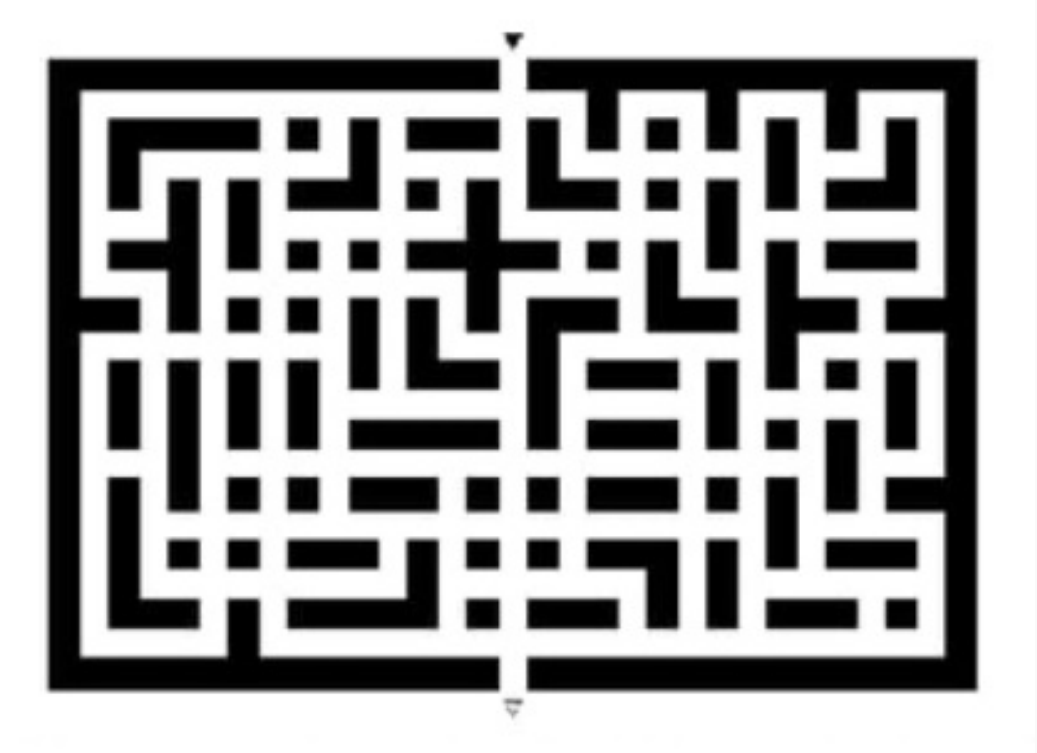


Figure 2: The bordered maze.

2) Determining coordinates of four corners of the maze.

Step 1: Creating a $(0,1)$ - matrix from the image of the maze

3) Shrinking the above $(0,1)$ - matrix into a smaller one (the desired matrix).
   We have two approaches:
   - Measure the width of the path.

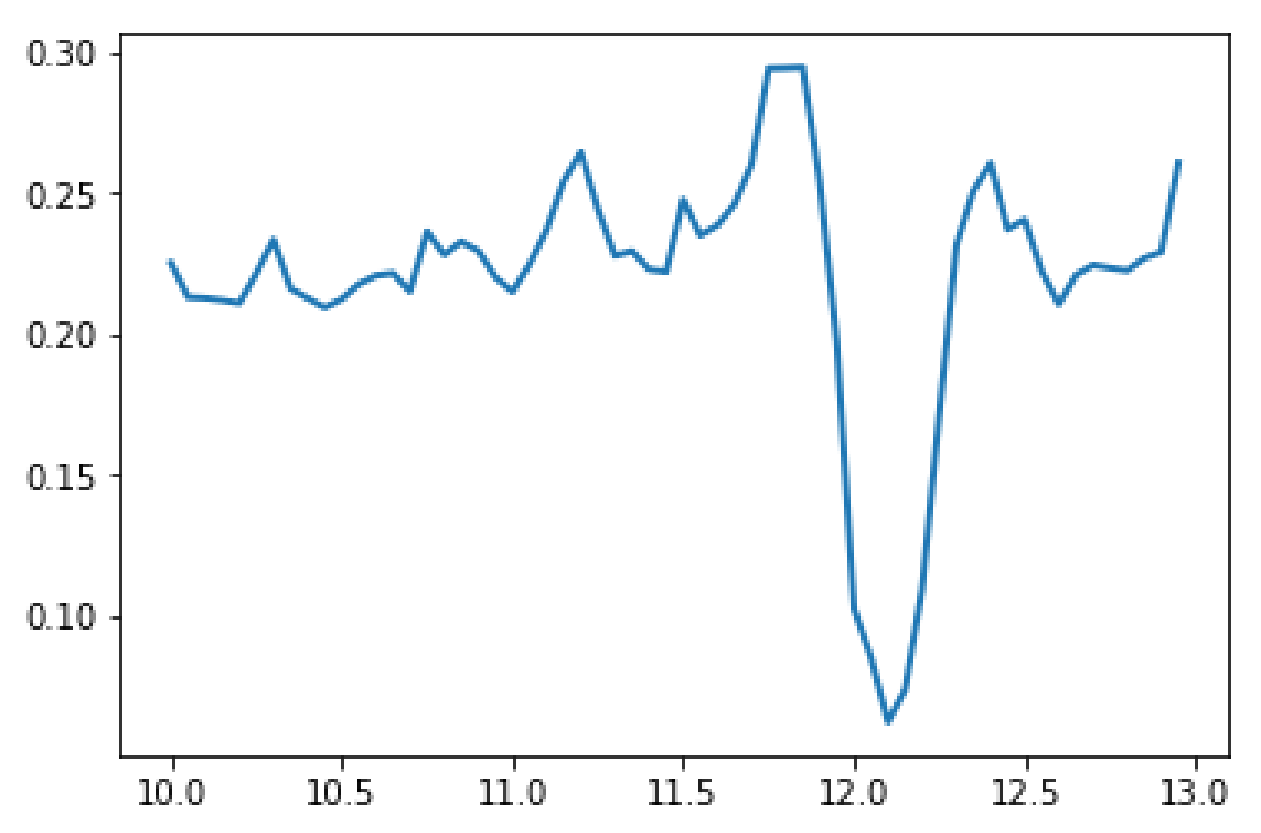


Figure 3: Error corresponding to the path's width.

Step 1: Creating a $(0, 1)$ - matrix from the image of the maze

- Determine the suitable size of the matrix.

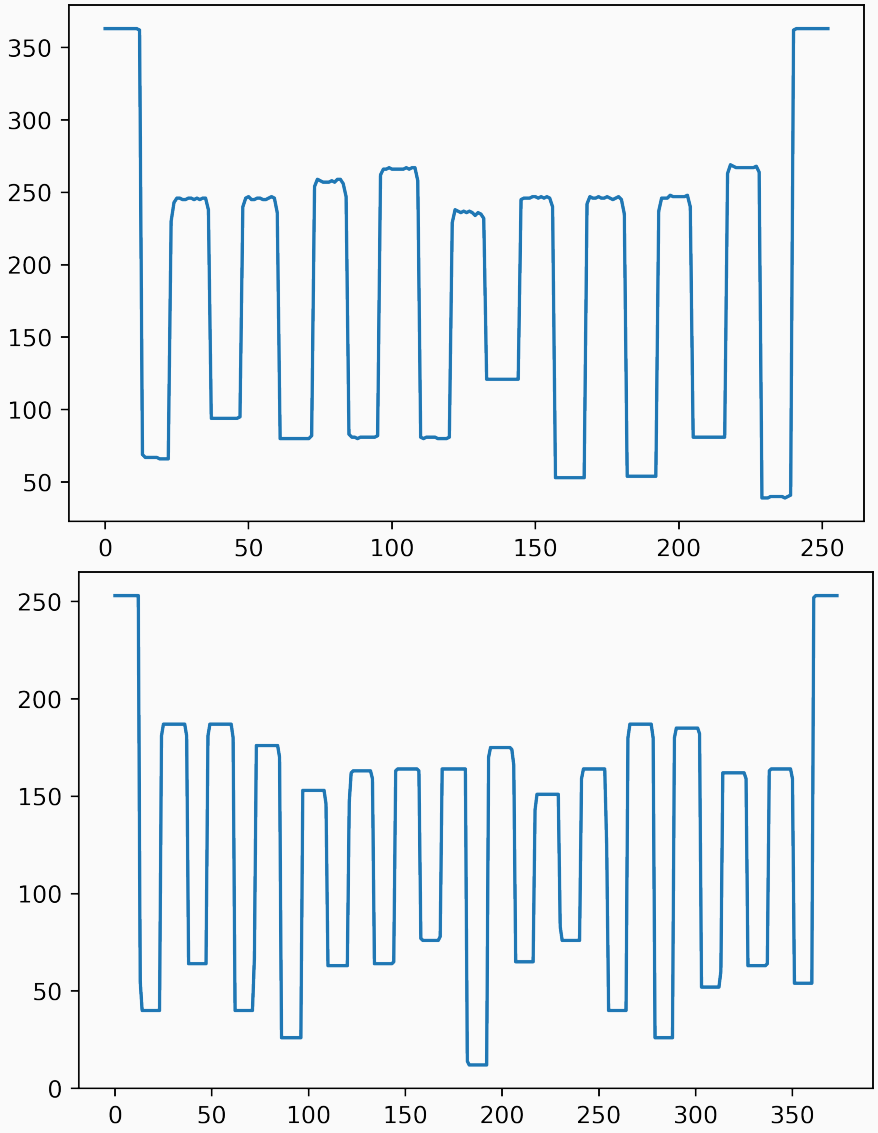


Figure 4: The horizontal sum and vertical sum.

Step 2: Changing matrix to a directed graph

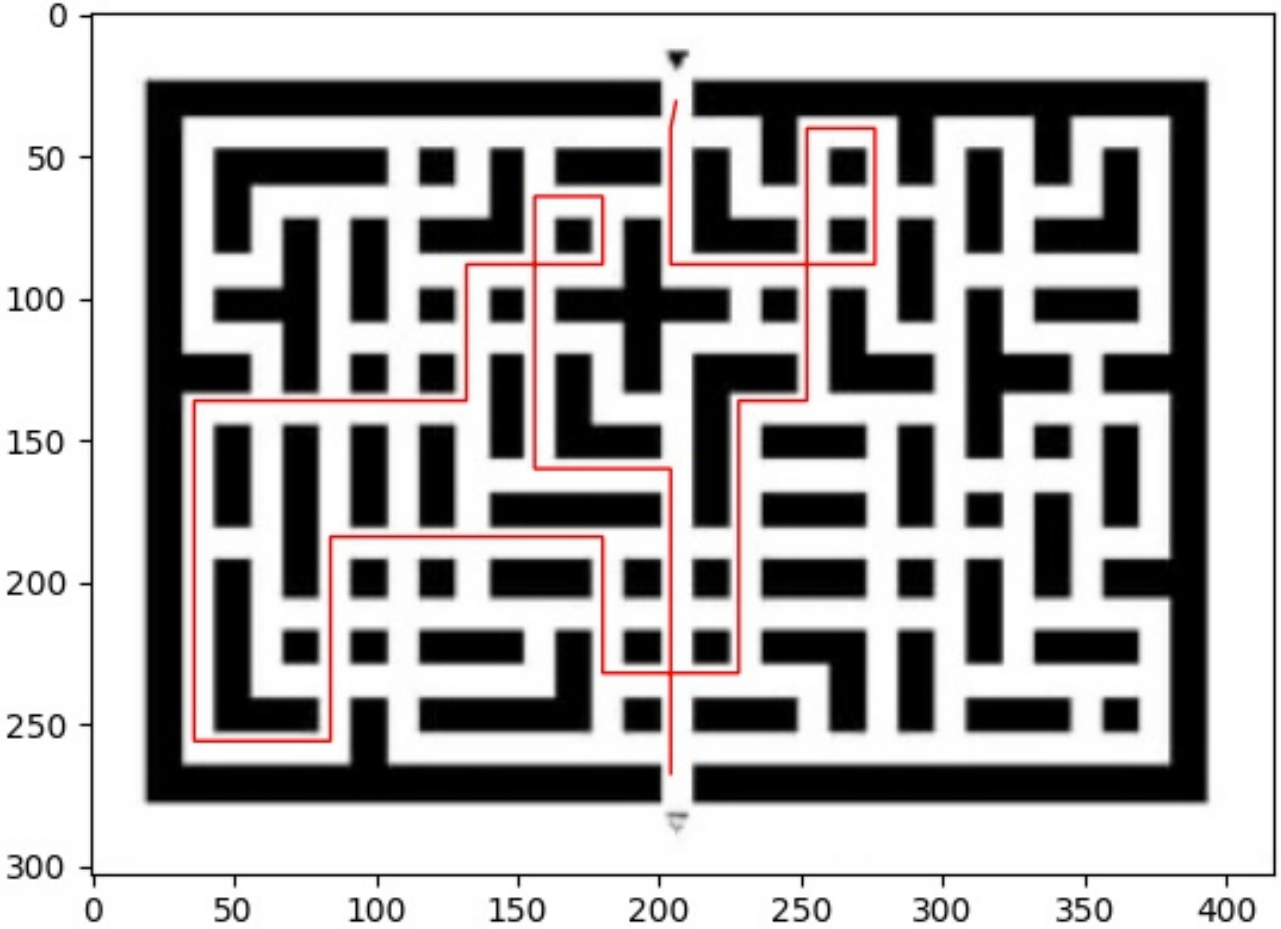
- Create a graph G.
- For each position which is empty, we add four vertices to the graph, which imply the position from the previous step.
- Add nodes to the graph which follow some rules.

Step 3: Find a path of the directed graph and visualize it

- Find a path of the directed graph which is output of Step 2.
- Visualize the path by using the matplotlib package in Python.

Results

Path in the maze



Statistics

- 379 lines of code.
- Commits counter:
  - Do Xuan Anh: 11
  - Le Quang Dung: 27
  - Hoang Anh Quan: 11
  - Trinh Huy Vu: 11

What we have learned

- Version control system: Git/Github.
- Packages in Python: matplotlib, networkx, numpy, PIL.
- Teamwork management.


Thank you!