**Maze-generator and Path-finder**

The program generates a maze and deploys various path finding algorithms on the maze and maps.

It also maintains a log of the path length found and time taken by the algorithms on various maze/maps.

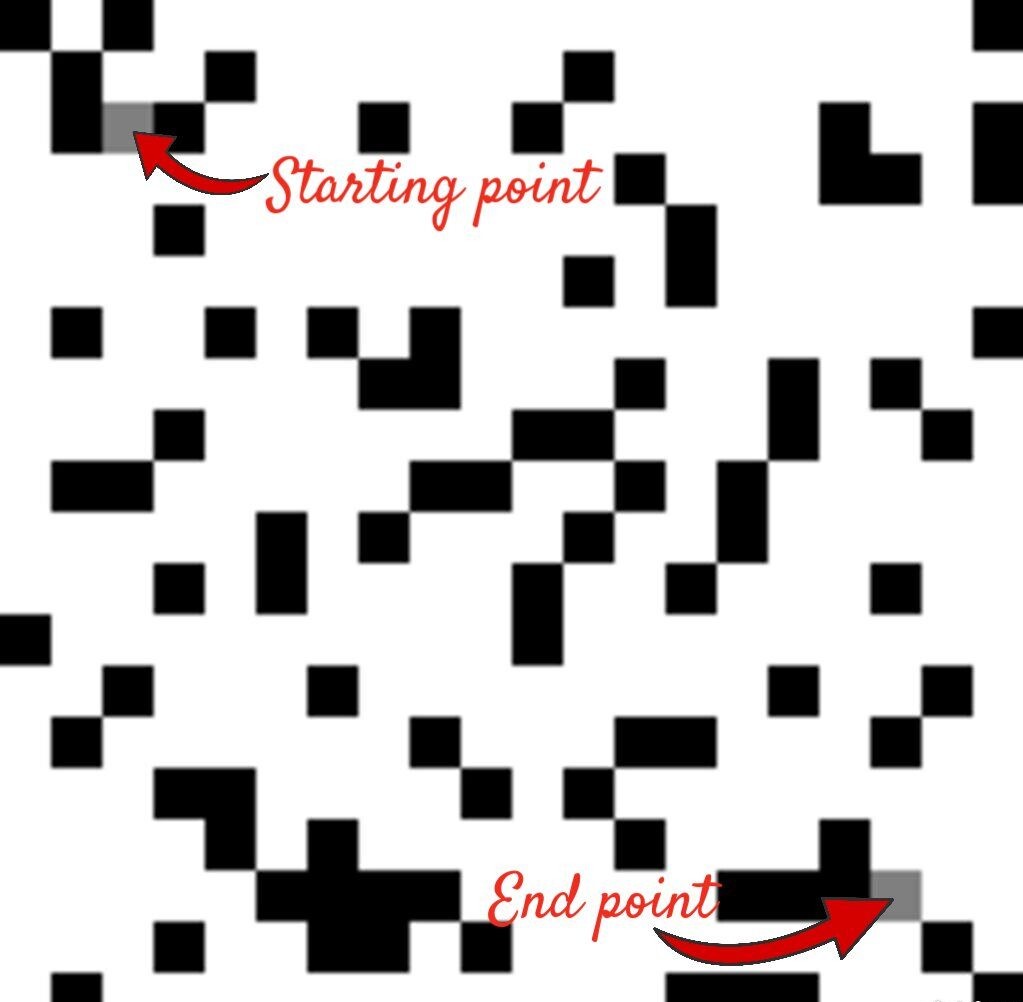
The project uses **python** along with numpy and opencv packages.

The algorithms used are :

* **Breadth-first search -** It explores equally in all directions and guarantees the shortest path.
* **Depth-first search -** Traverses by exploring as far as possible down each path before backtracking and does not guarantee the shortest path.
* **Best-first search (Greedy) -** It makes a choice based on the shortest heuristic distance from the goal node. It does not guarantee the shortest path.
* **Dijkstra’s algorithm -** Instead of exploring all paths equally, it favours lower cost paths and also, does guarantee the shortest path.
* **A\* algorithm -** It is a modification of Dijkstra’s Algorithm that is optimised for a single destination. It prioritises paths that seem to be leading closer to a goal.

Note : To close any image or to exit any path finding process midway press “q” on keyboard.

On every run, the program generates a new maze, which typically look like this



Also, there are two maps available to deploy path-finding algorithms on -

Map - 1



Map - 2



Setup

* Download the folder ‘Task1’
* After opening the ‘Task1’ folder as your **current working directory** in the terminal, run

pip3 install -r requirements.txt

or if you have pip installed, instead of pip3, then run

pip install -r requirements.txt

to install numpy and opencv-python packages required

* Run ‘runner.py’ to start the program