Student name: Nguyễn Khánh Toàn

Student number: 22127418

Mid term

Resources
- **Algorithm:**
  1. **ChatGPT**
  2. **My available project : https://github.com/KTNguyen04/SearchyVisualiz ation.git**
- **Visualization:**
  1. **My available project : https://github.com/KTNguyen04/SearchyVisualiz ation.git**
- **Tools:**
  1. **IDE: VisualStudioCode with environment**
  2. **Language: Python with Pygame**
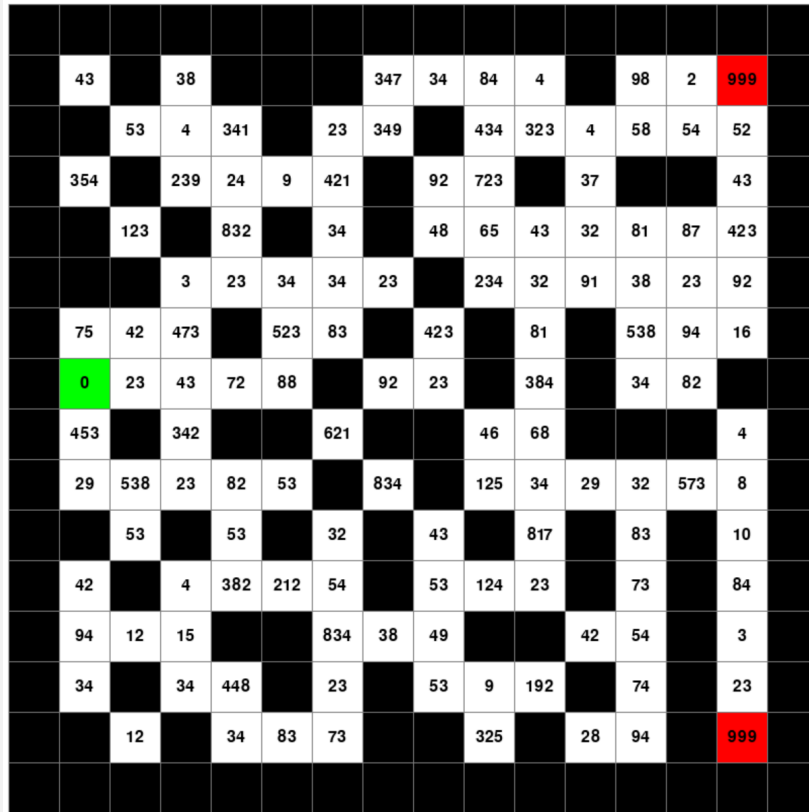- **Source for this project:**

Q1

a)

Figure 1: Maze


```
1    -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
2    -1 43 -1 38 -1 -1 -1 347 34 84 4 -1 98 2 999 -1
3    -1 -1 53 4 341 -1 23 349 -1 434 323 4 58 54 52 -1
4    -1 354 -1 239 24 9 421 -1 92 723 -1 37 -1 -1 43 -1
5    -1 -1 123 -1 832 -1 34 -1 48 65 43 32 81 87 423 -1
6    -1 -1 -1 3 23 34 34 23 -1 234 32 91 38 23 92 -1
7    -1 75 42 473 -1 523 83 -1 423 -1 81 -1 538 94 16 -1
8    -1 0 23 43 72 88 -1 92 23 -1 384 -1 34 82 -1 -1
9    -1 453 -1 342 -1 -1 621 -1 -1 46 68 -1 -1 -1 4 -1
10   -1 29 538 23 82 53 -1 834 -1 125 34 29 32 573 8 -1
11   -1 -1 53 -1 53 -1 32 -1 43 -1 817 -1 83 -1 10 -1
12   -1 42 -1 4 382 212 54 -1 53 124 23 -1 73 -1 84 -1
13   -1 94 12 15 -1 -1 834 38 49 -1 -1 42 54 -1 3 -1
14   -1 34 -1 34 448 -1 23 -1 53 9 192 -1 74 -1 23 -1
15   -1 -1 12 -1 34 83 73 -1 -1 325 -1 28 94 -1 999 -1
16   -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Figure 2: Matrix input in file cost.txt
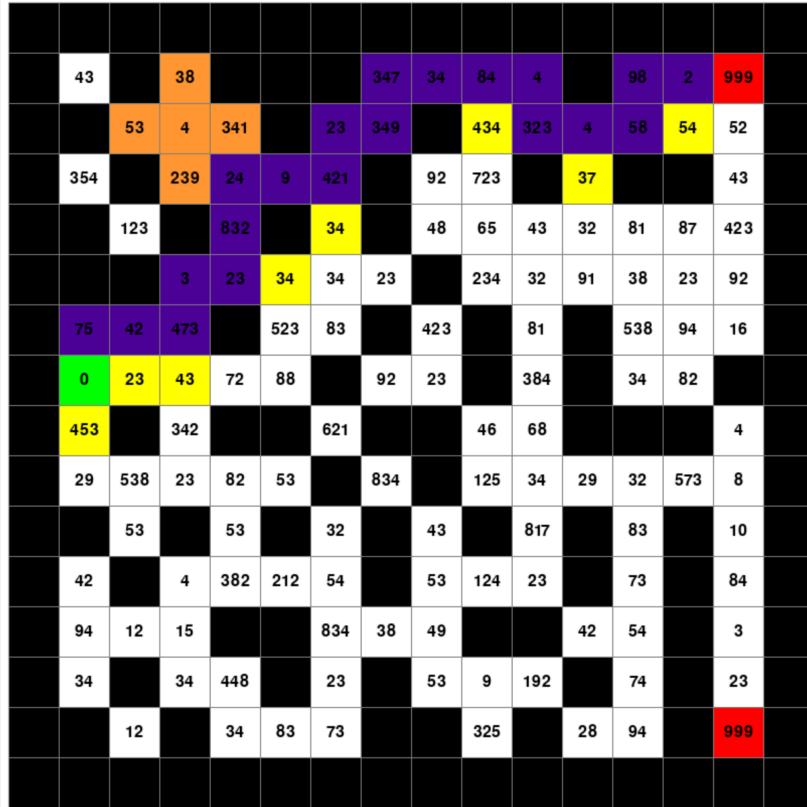
b)

c)

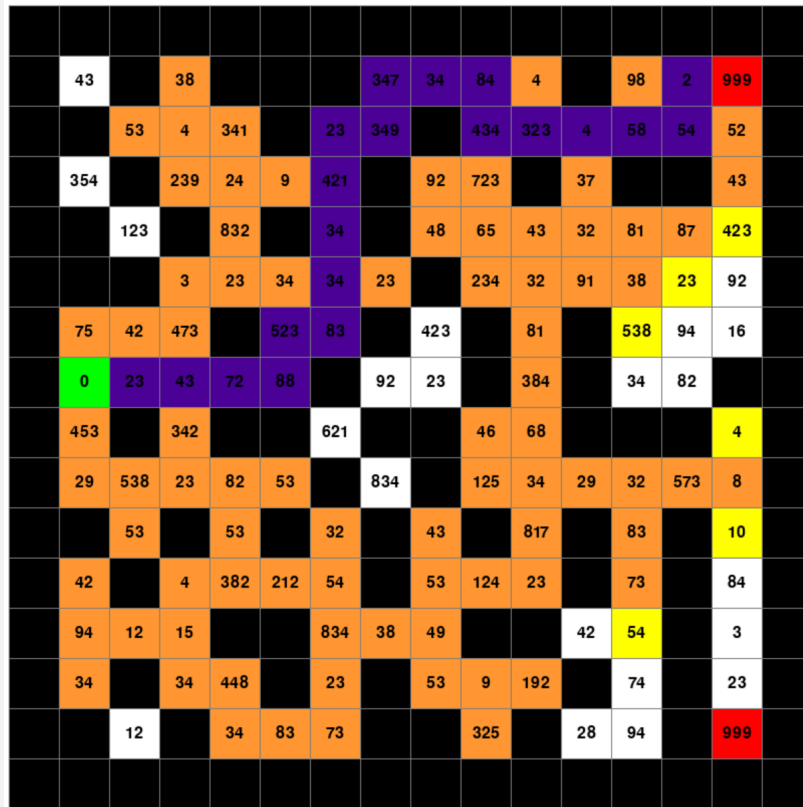**d)**

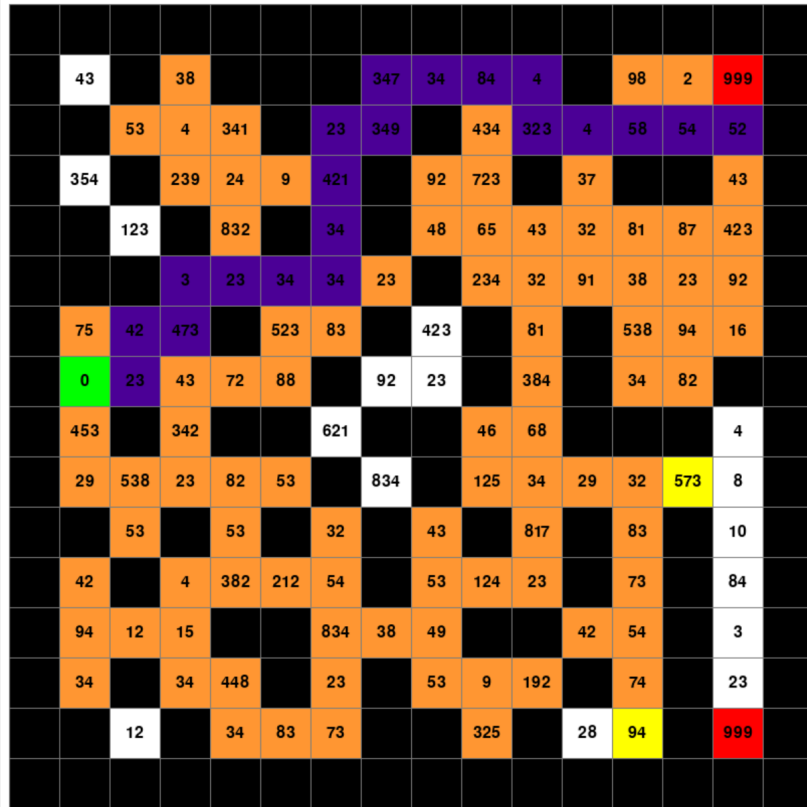# Q2. DFS ,BFS, AStar

a)

Figure 3: DFS

b)

Figure 4: BFS

c)

Figure 5: AStar

d)

Compare			
	DFS	BFS	A*
Time	13.68*C	39.15*C	42.30*C

Figure 6: Executed time

Q3. Hill-climbing search

a)

My implement for Hill-climbing search

```python
def hill_climbing(self, drawer) -> None:
    current_cell: None = self.start
    while True:
        self.select(cell=current_cell)
        self.drawy(drawer=drawer)

        if self.check_goals(cell=current_cell, drawer=drawer):
            self.drawy(drawer=drawer)
            return

        next_best_cell = None
        for neighbor_cell in self.get_neighboor(cell=current_cell):
            if (self.is_free(cell=neighbor_cell) or neighbor_cell.color == SELECTED_COLOR) and self.heuristic(cell=neighbor_cell) <= self.heuristic(cell=current_cell):
                next_best_cell: Any = neighbor_cell
                break

        if next_best_cell is None:
            break

        self.expand(cell=next_best_cell)
        current_cell: Any = next_best_cell

    self.reached(cell=current_cell)
    self.drawy(drawer=drawer)
```

Figure 7: Hill-climbing search implementation

```python
def stochastic_hill_climbing(self, drawer) -> None:
    current_cell: None = self.start
    while True:
        self.select(cell=current_cell)
        self.drawy(drawer=drawer)

        if self.check_goals(cell=current_cell, drawer=drawer):
            self.drawy(drawer=drawer)
            return

        next_best_cells: list[Any] = []
        max_heuristic = float(__x='-inf')
        for neighbor_cell in self.get_neighboor(cell=current_cell):
            if (self.is_free(cell=neighbor_cell) or neighbor_cell.color == SELECTED_COLOR):
                neighbor_heuristic: Literal[10000000000] = self.heuristic(cell=neighbor_cell)

                if neighbor_heuristic > max_heuristic:
                    next_best_cells: list[Any] = [neighbor_cell]
                    max_heuristic = neighbor_heuristic
                elif neighbor_heuristic == max_heuristic:
                    next_best_cells.append(neighbor_cell)

        if not next_best_cells:
            break

        next_cell: Any = random.choice(seq=next_best_cells)
        self.expand(cell=next_cell)
        current_cell: Any = next_cell

    self.reached(cell=current_cell)
    self.drawy(drawer=drawer)
```

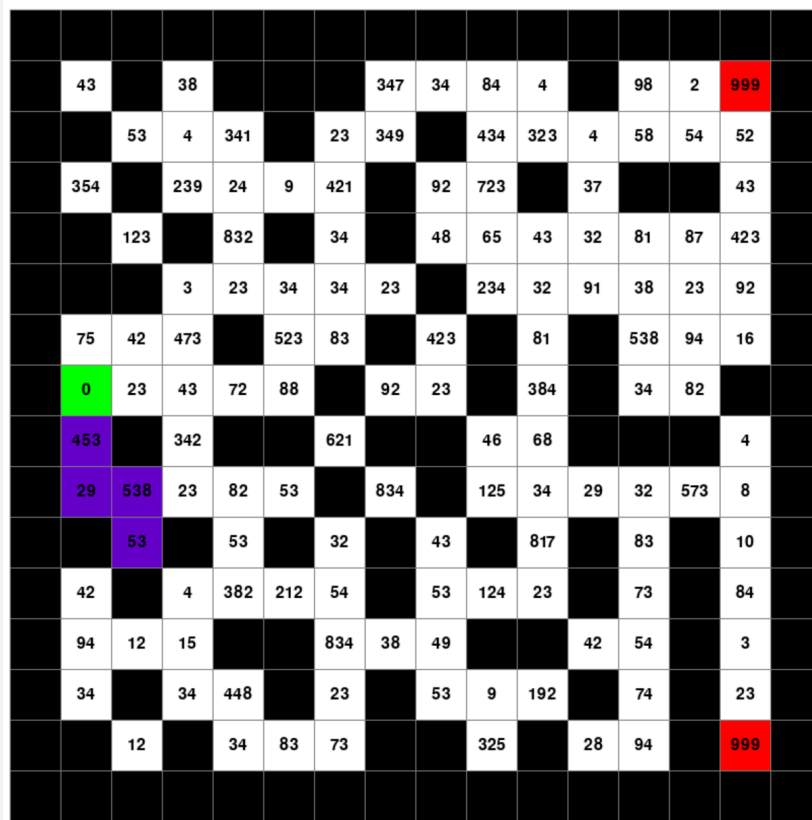Figure 8: Stochastic Hill-climbing search implementation

b)

Hill-climbing search

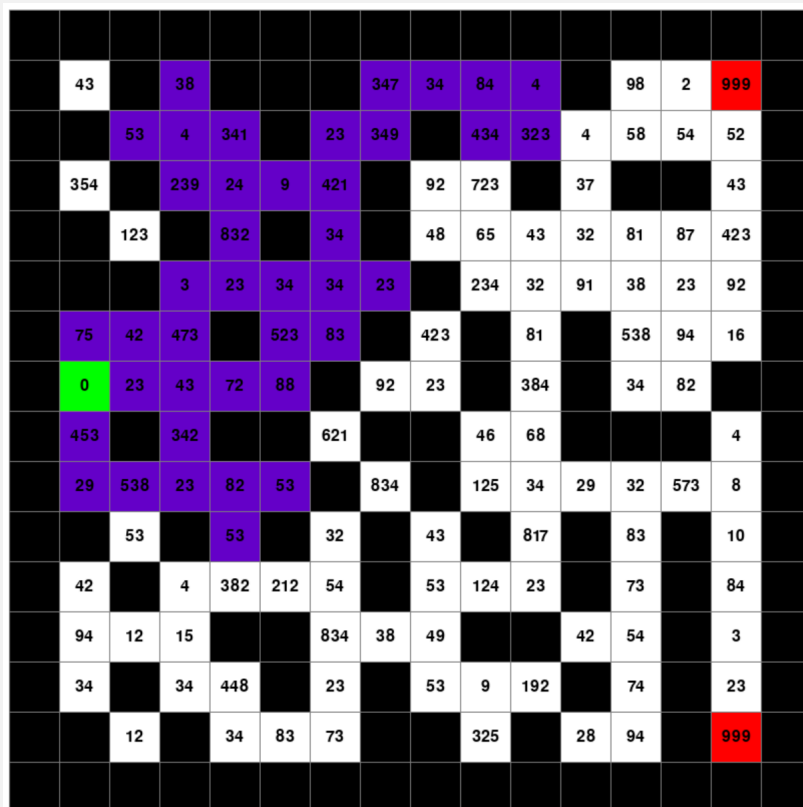Figure 9: Pure Hill-climbing search which get stuck in my maze

Figure 10: Stochastic Hill-climbing search which take more than 5min in my maze

c)

Compare

	DFS	BFS	A*	Hill-climbing	Stochastic Hill-climbing
Time	13.68*C	39.15*C	42.30*C	Stuck	>300*C

Figure 11: Searched time