Lab 1: Search

Subject: Fundamentals of Artificial Intelligence

Lecture: Pham Trong Nghia

TA: Nguyen Thai Vu

1. Introduction

- In this project, students research and implement the searching algorithm. In addition, students have to visualize the result of the searching algorithm.

2. Requirements

- Individual project.
- Programming language: Python (for visualization, we recommend students use turtle library of Python)
- Timeline: 2 weeks.
- Final product: student_ID.zip or student_ID.rar, includes:
    - Code folder: include every coding file.
    - Report folder: include file report.pdf:
        - Student's information
        - Each algorithm, student report:
            - The idea of the algorithm.
            - Example (reference section input/output)
            - Conclusion, pros, and cones.
- Evaluation:
    - Implement 5 searching algorithms: 70%.
    - Report: 30%
- Every cheat/copy/lie will be punished with a course score of 0.

3. Problem

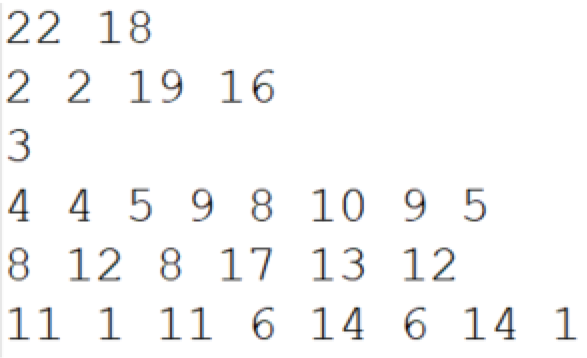
a. Problem description

- The robot has been sent to a maze of size M x N, and the robot has to find the path from the Source (starting position) to the Goal (ending position). The robot allows to move in 4 directions: up, down, left, and right. In the maze, there are some obstacles.
- The student was asked to implement 5 search algorithms:
    - **Breadth-first search**
    - **Uniform-cost search**
    - **Iterative deepening search** that uses depth-first tree search as a core component and avoids loops by checking a new node against the current path.
    - **Greedy-best first search** using the Manhattan distance as a heuristic.
    - **Graph-search A*** uses the Manhattan distance as a heuristic.

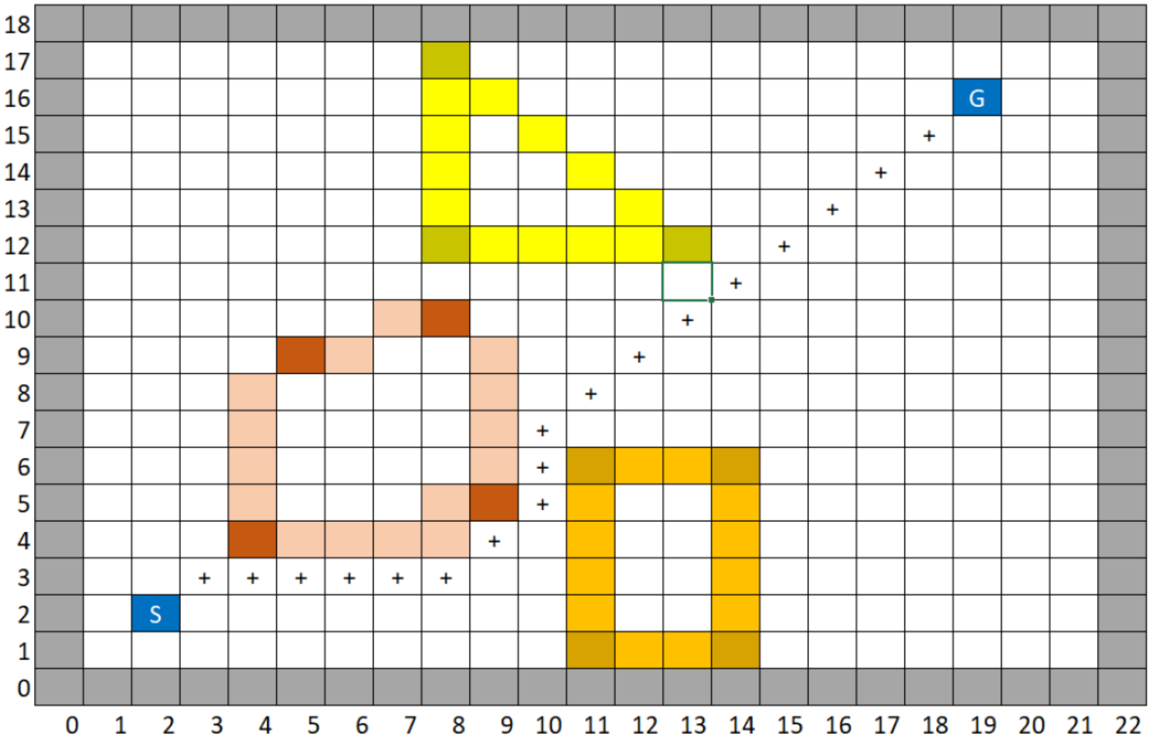
b. Input/output format

- The format of the input file:
    - First line: the size of the maze width, height.
    - Second line: the position of the Source and Goal. For example: 2 2 19 16 meaning source point is (2, 2) and goal point is (19, 16).
    - Third line: the number of obstacles in the maze.
    - The next following line defines the obstacle by the rule:
        - The obstacle is a Convex polygon.
        - A polygon is a set of points that are next to each other clockwise. The last point will be implicitly concatenated with the first point to form a valid convex polygon.
- The output:
    - Graphical representation of polygons.
    - Representation of expanded node.
    - Cost of the expanded node. (cost is 1 at every step)
    - Representation of path from source to goal.
    - Cost of the path. (cost is 1 at every step)
- The example of input.txt

  (Everything is relative, depending on your implementation)



```
22 18
2 2 19 16
3
4 4 5 9 8 10 9 5
8 12 8 17 13 12
11 1 11 6 14 6 14 1
```



4. References

- The document in the Computer Science Department at the University of Science, Vietnam National University, Ho Chi Minh City.