

Course: Algorithms Design and Analysis - CSC311

Semester: Fall 2023

Instructors: Prof. Mohamed Maher Ben Ismail & Prof. Aqil Azmi

Optimal Path Finding Using Dynamic Programming

DEADLINE: November 23rd 2023, 23:59.

1. Project Description

This project will help students understand how dynamic programming can be used to solve complex problems efficiently by breaking them down into simpler subproblems and reusing solutions to these subproblems. Specifically, the goal of this project is to implement a path finding algorithm using dynamic programming principles. The algorithm should be able to find the optimal path from a start point to an end point in a grid-based environment with obstacles.

- **Grid Environment:** Create a 2D grid of size $n \times m$ where each cell represents a location in the environment. Some cells are marked as obstacles and are impassable.
- **Start and End Points:** Choose two distinct cells in the grid as the start and end points.
- **Path Cost:** Assign a cost for moving from one cell to another. The cost can be the Euclidean distance between the centers of the two cells.
- **Dynamic Programming Algorithm:** Implement a dynamic programming algorithm that finds the path from the start point to the end point with the minimum total cost. The algorithm should avoid obstacles.
- **Path Visualization:** Visualize the optimal path on the grid.

NB: You will be working in a team of **two** students to fulfill the project requirements.

2. Programming

Implementation of algorithms may be done in any language of student's choice. However, the language and its compiler should support certain features in order to be able to run the experiments properly. The choice of C, C++, Java, Python, Matlab or the like should be enough. Source code has to be handed over.

3. Project Demonstration

Once the project is completed, the following is expected from you:

- A demonstration of your project in which you show the various features of your system such as its correctness, efficiency, etc. You should be prepared to answer detailed questions on the system design and implementation during this demo. We will also examine your code to check for code quality, code documentation, etc.
- You should hand in a completed project report which contains details about your project, such as main data structures, main components of the algorithm, design of the user-interface for input/output, experimental results, e.g. charts of running time versus input size, etc.
- You should also turn in your code and associated documentation (e.g. README files) so that everything can be backed up for future reference.
- Email your code and all associated files with “CSC311-Project <Lastname>” as subject.

4. Deliverables

- Source code of the implemented algorithm.
- A report explaining how the algorithm works, how dynamic programming principles are applied, and an analysis of its time and space complexity.
- Test cases demonstrating that the algorithm works correctly, including edge cases such as when there is no possible path due to obstacles.

5. Report

A report describing the following points must be handed over.

- Explanation of the algorithms.
- Brief explanation of the implementations. It can be done by including sufficiently detailed comments in the code.
- User-interface for input/output, experimental results
- Discussion and analysis of the obtained results.
- Conclusions. In this section students must draw their own conclusions (be creative).

The report has to be written in correct English; it also has to possess clarity of thought.

6. Evaluation Criteria

- Correctness of the implemented algorithm.
- Quality of the report, including clarity of explanations and thoroughness of complexity analysis.
- Robustness against different test cases.
- Code quality, including readability, modularity, and use of appropriate data structures.

7. Questions and Office Hours

Mr. Wesam Hatamleh, Prof. Aqil Azmi and Prof. Mohamed Maher ben Ismail are willing to answer your questions about algorithms, complexity or the experiment. They will not answer questions about coding errors as it is our feeling that, at this point, writing error-free code is your responsibility.