

A Level Programming Project Report

Junrong Chen

May 24, 2021

Contents

1	Analysis	2
1.1	Description of the problem	2
1.2	Stakeholders	2
1.2.1	Computer Science Teachers	2
2	Design	4
3	Development	5
3.1	Preparation	5
3.1.1	Code Editor	5
3.1.2	Runtime environment	5
3.1.3	Version control	5
3.1.4	Project management	6
4	Evaluation	7

Chapter 1

Analysis

1.1 Description of the problem

A Level Computer Science students are required to learn many algorithms and data structures. In an actual exam, they are asked to handwriting the algorithm to solve some problems. Many students find it is hard to achieve high score in those type of questions due to the lack of efficient training. When the students learn and practice the algorithms, the general procedure is that they are given some questions from the teacher, then they attempt to solve them by themselves, and finally they self mark their solution based on the mark scheme. This procedure works well for general questions. However, for the algorithm questions, different students will produce completely different code solutions, which makes the self marking very unreliable. It is also too much work for the teacher to mark the solution one by one. So in the end, students do not know whether they get things right and teachers do not know how the students perform and how they can help - especially in this lockdown online learning era where no direct contact between teacher and students is possible.

Both the students and the teachers are looking for a more efficient method to learn and practice.

1.2 Stakeholders

There are two types of stakeholders, Computer Science teachers and Computer Science students.

1.2.1 Computer Science Teachers

Computer Science teachers find it is difficult to monitor their students' understanding of the coding and algorithms part of the subject, so they cannot provide sufficient help to their students. This software allows them to create such type of coding questions and send them to the students. After the students

hand their solutions back, the software will automatically mark their answers and provide detailed statistical data with visualisations. This helps the teachers saving a lot of time and allows them to help the students better.

Chapter 2

Design

This is the first section.

Chapter 3

Development

3.1 Preparation

3.1.1 Code Editor

Download and install VS Code.

Instead of using a large IDE with everything pre-configured, I decide to use a code editor to write the code and a terminal to execute all the commands I need. This gives me more control on my project.

VS Code is a free and open source code editor which also have greate support for Python.

I decide to use VS Code as my main editor.

3.1.2 Runtime environment

Download and install Python 3.9.5.

I simply choose the latest Python release for this project. When we release the software, the Python interpreter will be packed with the binary files so there is no need to worry about the compatibility with the Python installation in users' environment.

3.1.3 Version control

Download and install Git.

Git is a free and open source distributed version control system. It records every "commit" I made to the source code and allows me to revert back to any previous "commit". This makes it easy to roll back to a certain version and locate bugs. It also allow me to create new "branches" which is useful when experimenting new features without worrying about damaging the stable code.

I decide to use Git as the version control system for this project.

3.1.4 Project management

GitHub

GitHub is a code hosting platform which supports many project management features. The "Issue" allows my stakeholders to report bugs and suggests features. The "Action" provides support for CI/CD. The "Project" provides support for manage and organize the TODO list for the project.

I decide to use GitHub as the code hosting platform for this project.

```
import numpy as np

def incmatrix(genl1,genl2):
    m = len(genl1)
    n = len(genl2)
    M = None #to become the incidence matrix
    VT = np.zeros((n*m,1), int) #dummy variable

    #compute the bitwise xor matrix
    M1 = bitxormatrix(genl1)
    M2 = np.triu(bitxormatrix(genl2),1)

    for i in range(m-1):
        for j in range(i+1, m):
            [r,c] = np.where(M2 == M1[i,j])
            for k in range(len(r)):
                VT[(i)*n + r[k]] = 1;
                VT[(i)*n + c[k]] = 1;
                VT[(j)*n + r[k]] = 1;
                VT[(j)*n + c[k]] = 1;

            if M is None:
                M = np.copy(VT)
            else:
                M = np.concatenate((M, VT), 1)

            VT = np.zeros((n*m,1), int)

    return M
```

Chapter 4

Evaluation

This is the first section.