A Level Programming Project Report

Junrong Chen

November 2, 2021

Contents

1 Analysis **3**

1.1 Problem Identification 3

1.2 Stakeholders 3

1.2.1 Computer Science teachers 3

1.2.2 Computer Science students 4

1.3 Why it is suited to a computational solution 4

1.4 Solve by computational methods 4

1.4.1 Thinking abstractly 4

1.4.2 Thinking ahead 5

1.4.3 Thinking procedurally and decomposition 5

1.4.4 Thinking concurrently 6

1.5 Interview 6

1.5.1 Design interview 6

1.5.2 Conduct the interview 7

1.6 Research 10

1.6.1 LeetCode 10

1.6.2 Codeforces 15

1.7 Features 17

1.8 Limitations 18

1.9 Hardware and software requirements 20

1.10 Success criteria 21

2 Design **23**

2.1 Decomposition 23

2.1.1 Navigation View 25

2.1.2 HomePage 26

2.1.3 Problems Page 27

2.1.4 AssignmentsPage 28

2.1.5 Playground 29

2.1.6 Account Page 30

2.1.7 Settings Page 31

2.1.8 Create New Problem Page 32

2.1.9 Create Assignment Page 33

2.1.10 Judger Module 33

2.2 Algorithm design 33

2.3 Data structure design 33

2.3.1 Class design 33

3 Development **39**

4 Evaluation **40**

Chapter 1

Analysis

1.1 Problem Identification

A Level Computer Science students need to learn many algorithms and data structures during the course. In the final exam, they need to write pseudocode to solve computational questions. Many students find it is hard to achieve a high score on those questions due to the lack of efficient training. The general method used by students to learn and revise for Computer Science is to attempt and self-mark past paper questions. This works well for ordinary questions. However, for the algorithm questions, different students may produce completely different code solutions. This makes their self-marking very unreliable. It is also too much work for the teacher to mark their solutions 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 teachers 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' ability to design and implement algorithms, so they cannot provide efficient help to

their students. This software allows them to create 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 simple visualizations. This helps the teachers saving a lot of time and allows them to help the students better.

The stakeholder is Mr Grimwood, who is an experienced A Level Computer Science teacher who teaches a Year 12 CS group and a Year 13 CS group.

1.2.2 Computer Science students

Computer Science students find that they tend to lose marks on the algorithms coding questions, so they want more practice. But unlike ordinary questions, they may take a completely different approach towards the questions comparing to the mark scheme, so they do not know whether they get it correct. Students may also think they have got things right, but actually, they have made some mistakes. The software provides a free practice space that automatically marks their solutions and points out their mistakes in real-time. So the students can learn and revise more efficiently.

The stakeholders are Timofei and PCloud. They are both Year 13 students studying A Level Computer Science.

1.3 Why it is suited to a computational solution

The original problem, 'understand and mark a student's answer' is a very difficult question for a computer to solve. But I transform the question into 'compare the output of the students' code with pre-generated test cases', which makes the problem solvable using a computational method since a computer is good at 'executing a piece of code' and 'comparing two strings'. This approach solves the 'marking' question from another angle and makes the question suited to a computational solution.

1.4 Solve by computational methods

1.4.1 Thinking abstractly

In reality, students use pens and paper to write their code solutions. This can be simplified into a code editor, and the students can use their keyboards to type in the code. In this way, no 'text scanning' or 'handwriting recognization' is needed which makes the design and programming much easier. The code editor will also provide a better user experience. Features such as syntax highlighting cannot exist on paper but are possible in a code editor.

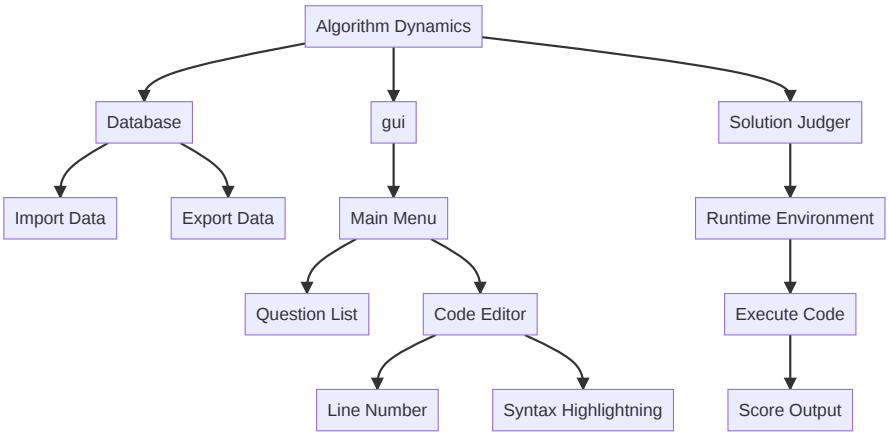
In reality, the students' answer is sent to a teacher to mark it against the mark scheme. The teacher needs to read the code line by line and check whether it is correct. This process is abstracted into a judger that marks the code against pre-generated test cases, which transforms a problem that originally cannot be solved by computational method into one which is very easy to be solved by a computer while saving time and costs. When creating a new question, instead of creating a mark scheme for marking, the teacher needs to provide test cases with the correct input and expected output. The judger will run the students' submissions with the input and check whether their output matches the expected one.

1.4.2 Thinking ahead

For teachers, the software requires them to enter questions and test cases. A question editor containing input boxes is needed for this purpose. For students, the software requires them to enter their code solutions. A code editor is needed for this purpose. A relational database is needed to store all the data. For all users, the software requires input data from the mouse and keyboard to navigate between different windows and menus. Users will also need a monitor for the program to display all the information and outputs.

1.4.3 Thinking procedurally and decomposition

The program can be decomposed into several parts. Each part can be designed and maintained individually. Different components can interact with each other using custom APIs.



1.4.4 Thinking concurrently

When judging the students' solution, many test cases can be executed at the same time to reduce the judging time. The number of parallel judgers needs to be set carefully based on the user's hardware. Running too few test cases concurrently may result in a very long judging time while running too many test cases at the same time may use up computing resources and cause issues.

1.5 Interview

1.5.1 Design interview

Interview for teachers

1. Do you find your students tend to lose marks on programming questions in exams?
2. Do you find marking the programming question takes a lot of time and effort?
3. Compare to the knowledge-based Computer System section, do you find it is more difficult to monitor students' skill level on the Algorithm and Programming section?
4. Have you ever heard about some online programming platforms?
5. Have you ever tried some of the online programming platforms?
6. If yes, what do you think about these platforms? Have you ever considered using them for teaching and training?
7. Do you think a similar solution can help improve the efficiency of learning and training?
8. If no, do you think the idea of a software that can mark students' answers on programming questions and provide analysis data can help improve the efficiency of learning and training?
9. Do you have anything else to add?

Question 1 to 3 is a series of proof-of-concept questions, which I expect my stakeholders to answer 'Yes' to all of them. They confirm that the problem I am trying to solve exists and there is a need for such a solution. Question 4 to 5 asks about the teachers' knowledge of existing solutions. Question 6 to 8 ask about their experiences and opinions about these existing solutions, which gives me insights on the problems with existing solutions and how my solution can fit their need better.

Interview for students

1. Do you find the programming questions difficult?
2. Do you find yourself lacking efficient practising in algorithm designing and programming?
3. Have you ever heard about some online programming platforms?
4. Have you ever tried some of the online programming platforms?
5. If yes, what do you think about these platforms?
6. Do you think a similar solution can help you learning and practising?
7. If no, do you think the idea of software that provides coding questions and marks your answer instantly can help you learn and practice better?
8. Do you have anything else to add?

Question 1 and 2 are similar proof-of-concept questions to confirm such a problem exists. The following questions ask about students' knowledge of existing solutions. If they have used an existing product before, I ask whether they think it helps. Otherwise, I ask whether they think it will be useful.

1.5.2 Conduct the interview

Computer Science teacher - Mr Grimwood

1. Do you find your students tend to lose marks on programming questions in exams?

   They do. Many of them don't understand the algorithms.

2. Do you find marking the programming questions takes a lot of time and effort?

   Yes. Because some students produce partially correct answers, so it takes a lot of time to identify the correct part and award them the corresponding mark. Some students may take completely different approaches which takes a lot of effort to understand and mark them.

3. Do you find it is more difficult to monitor students' skill level on the Algorithm and Programming section and more difficult to provide sufficient help?

   Yes.

4. Have you ever heard about some online programming platforms?

   I have. Emm... But I forget the names.

5. If yes, have you ever tried some of the online programming platforms?

   I have.

6. If yes, what do you think about these platforms?

   I think the idea is quite interesting and I find them working quite well.

7. Have you ever considered using them for teaching and practising?

   No. Because most of them require a paid subscription, and their content is more likely to be something like 'Learning Python' which is irrelevant to the A Level Computer Science content.

8. Do you think a similar solution can help improve the efficiency of learning and training?

   Yes. The students can learn at their own pace and they can keep practising by themselves.

9. Do you have anything else to add?

   No.

Mr Grimwood has several valuable points here. He points out that the 'partially correct' answers are the most difficult ones to mark. For my solution, if a student submits a 'partially correct' code answer, then its output will certainly not match the expected output. This means my solution might not be able to tell the difference between a 'partially correct' answer and an 'incorrect' answer. This is a potential limitation I need to watch out for. He also says the price is one of his concerns. My solution will be free and open-source, which will meet his need perfectly. By adding the function to create custom questions and share them with others, users will be able to create and find A Level Computer Science content, or any content easier. It is also a good idea for me to create some A Level Computer Science content that comes with the software to make it easier to use.

Computer Science student - PCloud

1. Do you find the programming questions difficult?

   I find some of them quite complex and difficult, especially the graph algorithms such as Dijkstra.

2. Do you find yourself lacking efficient practising in algorithm designing and programming?

   Absolutely. Although I code a lot in my spare time, normal projects are quite different from the exam questions. There are not many past papers and exam-style questions for practising, so I usually don't feel confident of those questions.

3. Have you ever heard about some online programming platforms?

   Yes. Such as AcWing, LeetCode, and TopCoder.

4. Have you ever tried some of the online programming platforms?

   Yes. I am an active user of AcWing.

5. If yes, what do you think about these platforms?

   I enjoy the experience. They can provide instant feedback for my submissions. It provides very strong positive feedback when I solve a new question. I find myself learning faster and more efficiently with such platforms.

6. Do you think a similar solution can help you learning and practising?

   Absolutely. The existing platforms do not provide A Level related content. So if a software solution can be altered for A Level Computer Science course, that will help a lot.

7. (*) How do you think it should be optimized for A Level CS content?

   You can add past exam questions practising. Add a timed practice mode will be helpful.

8. Do you have anything else to add?

   No.

PCloud confirms that such a solution will help him learning and practising more efficiently. The instant feedback of whether he gets the question correctly is very important to him. Instead of sending the user's submission to a remote server, my solution should judge the user's answer on their computer. This can avoid the instabilities caused by the remote server's availability and the network connection. He also gives me some good ideas about the content. I can add past exam questions for users to do timed practice, which enables users to practice algorithms and exam techniques at the same time.

Computer Science student - Timofei

1. Do you find the programming questions difficult?

   Yes. I generally lose marks because of some careless syntax mistakes I made.

2. Do you find yourself lacking efficient practising in algorithm designing and programming?

   Yes. I find I cannot find many materials to practice.

3. Have you ever heard about some online programming platforms?

   Codewar. Something like that.

4. Have you ever tried some of the online programming platforms?

   Yes.

5. If yes, what do you think about these platforms?

   I think they are quite helpful. But I find their marking is too specific, if I get a single character wrong in my output, it gets marked incorrect.

6. Do you think a similar solution can help you learning and practising?

   Yes.

7. Do you have anything else to add?

   No.

Timofei points out that the marking system in existing products is not very sensible. This may be a potential limitation of my solution as well. It is easy to directly compare the users' output and the expected output. But if they are different, it is difficult to figure out whether that difference is caused by a wrong code solution or just some formatting error. I can partially solve this by allowing the users to pre-test their code against examples before formal submissions, so they can check the output format.

1.6 Research

There are many coding training websites on the market, most of them share a similar idea, so I will investigate two of the most popular ones.

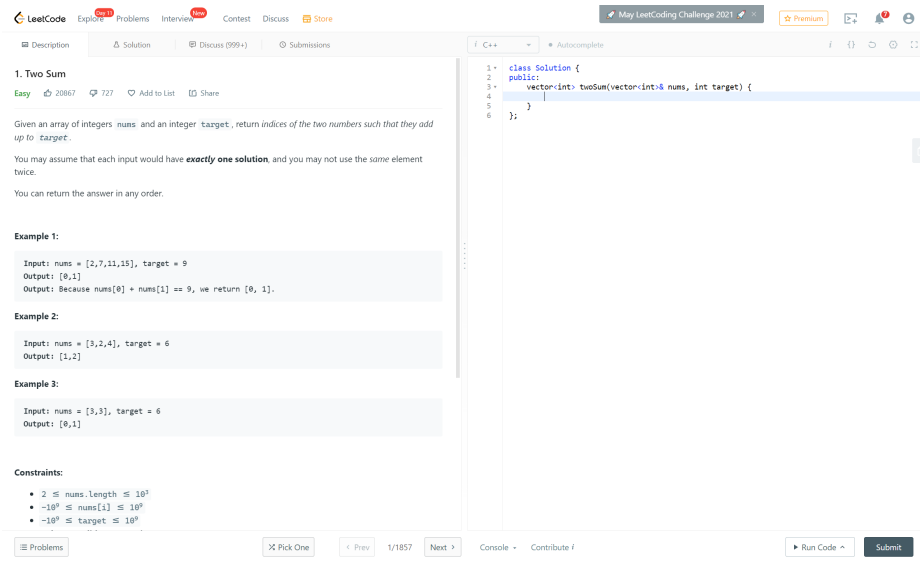
- LeetCode
- Codeforces

1.6.1 LeetCode

LeetCode is a platform for interview coding training, many large companies (Google, Facebook, ...) use it as a part of their interview.

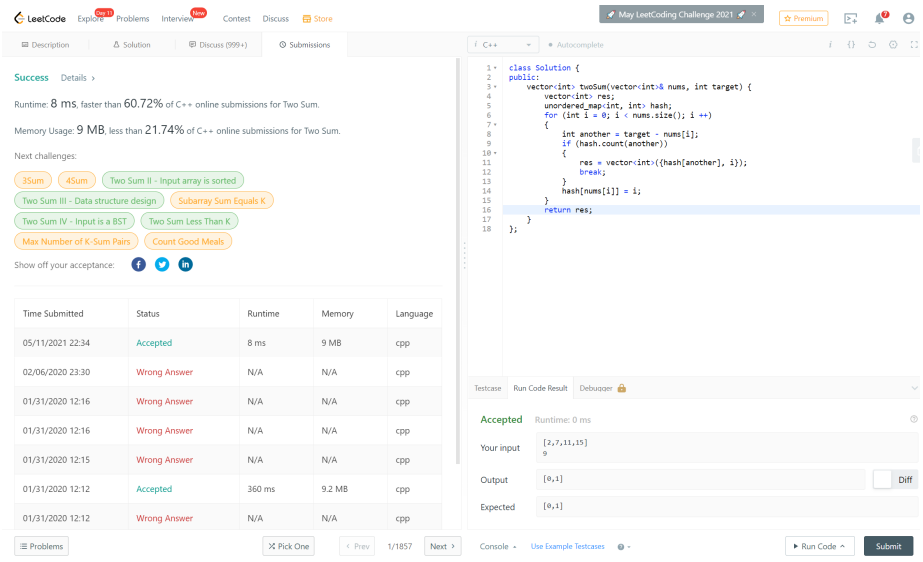
LeetCode provides a database containing more than 1000 coding questions.

Main coding layout



This is LeetCode's main coding area. The user's screen is split into two parts - the question section and the code editor for inputting answers. Users can drag the splitter in the middle to adjust the size of each section.

The question section contains 4 tabs, 'Description' tab displays the content of the question. 'Solution' tab displays the solutions from the community. 'Discuss' tab displays the discussions in the community. 'Submission' tab lists the user's previous submissions. Since I am not adding social functions in my solution, I will ignore the 'Solution' and 'Discuss' tabs. Under the 'Description' tab, LeetCode provides the context of the question, followed by 3 examples, and constraints for this question. The examples allow the user to run and check their solution before formal submission for marking, this can help them avoid silly mistakes. My solution should also provide similar examples for each question. Under the 'Submission' tab, LeetCode records every history submission, so the user can revise old questions more efficiently. My solution should provide a similar function as well.

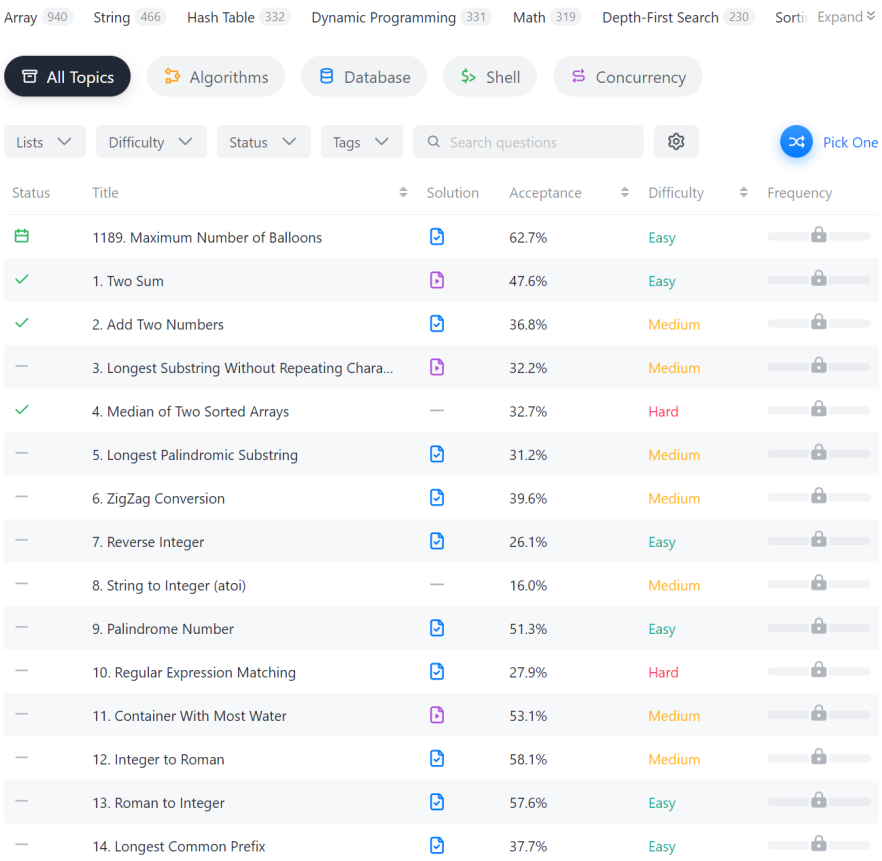


The code editor provides line number and syntax highlighting functions. User can change their programming language with a drop-down box. LeetCode supports all mainstream programming languages. My solution should be able to support multiple programming languages as well, which allows students with different backgrounds to use it easily.

On the button, the user can 'Run Code' to test their code against the examples before submission, and then click the 'Submit' button to submit their solution formally.

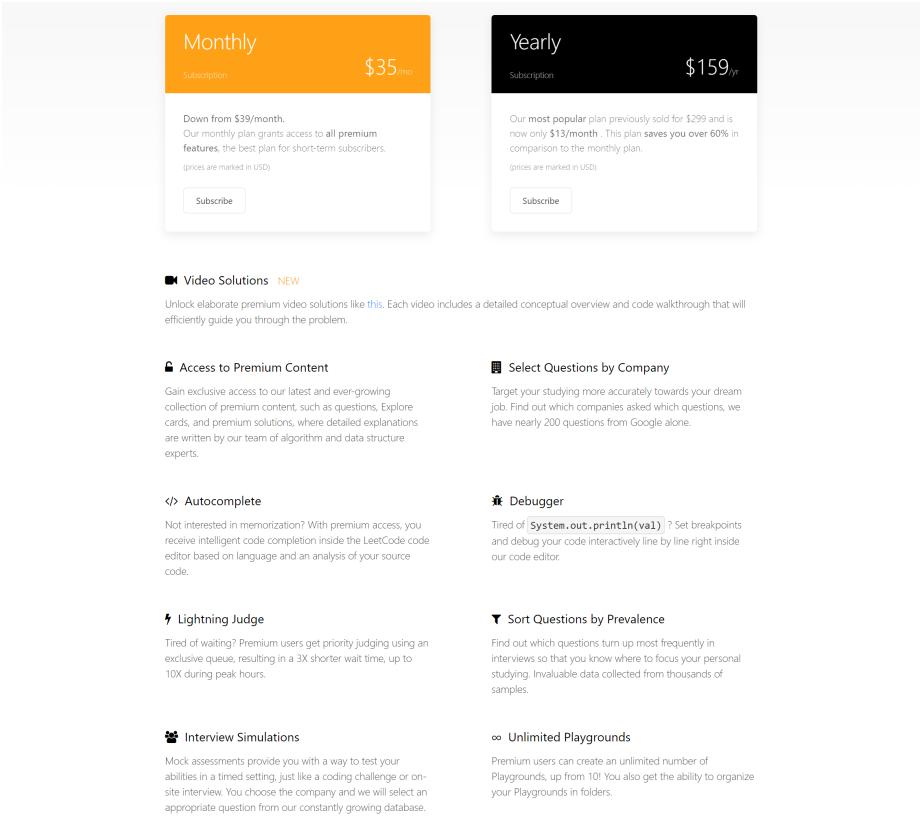
The split view design is clean and handy. The user can see the question and write their solution on the same page without switching between different windows. The design of examples and the 'Run Code' button is useful as well. I can refer to LeetCode's coding layout when designing my solution's interface.

Question database



Every question in LeetCode has many different attributes (Lists, Difficulty, Status, Tags, Title, Acceptance), so it is very easy for a user to find a suitable question to practice. My solution can similarly organize the question database and provide a corresponding query interface for a better user experience. The 'Pick One' button on the top right is a very handy feature as well. Users can simply click that button to start working on a quick random question. The idea of 'a list of questions' is great. Users can organize a series of questions to practice and share.

Pricing



The basic functions of LeetCode are free to use for all users and it charges a fee for premium subscriptions. The premium subscription provides a larger question database, better code editor, faster judger, and more.

Analysis

LeetCode is a fully web-based solution, which means it works on any device. However, it also means you will not be able to use it without a stable Internet connection. I decide to make my solution a desktop application since most students practice coding with a computer. It also save me a lot of cost from running and maintaining a server. LeetCode runs a large community for users to discuss questions with each other. I am not adding such a function to my solution. Teachers and students can use existing platforms they have been familiar with, it is unnecessary for me to develop a new platform and for the users to migrate from mature solutions. LeetCode has an easy-to-use graphical interface, which is important so new users can get their hands on very easily.

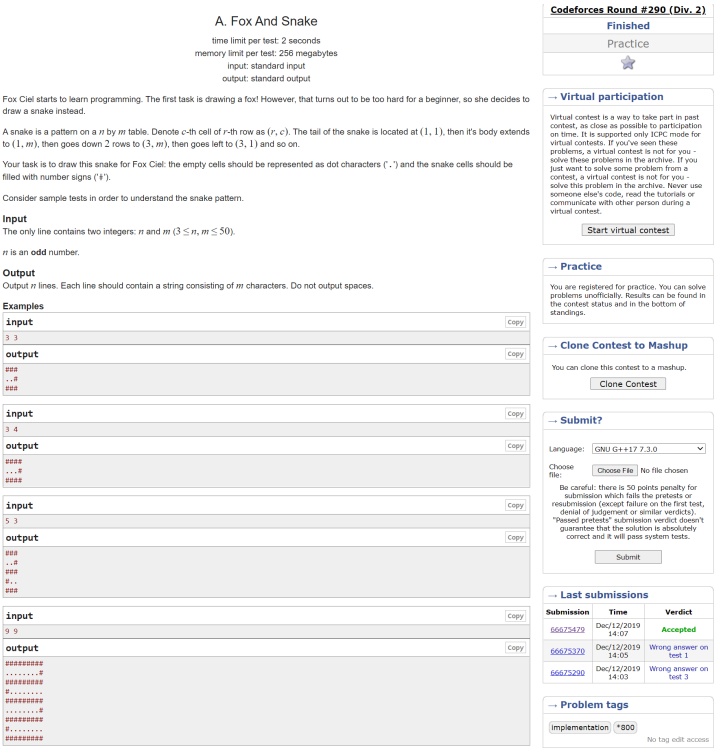
LeetCode does not support custom questions or any functions for educators. It is mainly designed for self-learners. My solution is designed for school use, so

it must support functions like custom questions, custom assignments, statistics data visualizations. LeetCode charges a subscription fee for essential functions. My solution will be free and open-source so everyone can benefit from it.

1.6.2 Codeforces

Codeforces is a competitive coding platform, it is mainly used by people to the held coding competition.

Main question layout



A. Fox And Snake

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Fox Ciel starts to learn programming. The first task is drawing a fox! However, that turns out to be too hard for a beginner, so she decides to draw a snake instead.

A snake is a pattern on a n by m table. Denote c-th cell of r-th row as (r, c). The tail of the snake is located at $(1, 1)$, then it's body extends to $(1, m)$, then goes down 2 rows to $(3, m)$, then goes left to $(3, 1)$ and so on.

Your task is to draw this snake for Fox Ciel: the empty cells should be represented as dot characters ('.') and the snake cells should be filled with number signs ('#').

Consider sample tests in order to understand the snake pattern.

Input
The only line contains two integers: n and m ($3 \le n, m \le 50$).

n is an **odd** number.

Output
Output n lines. Each line should contain a string consisting of m characters. Do not output spaces.

Examples

input
3 3
output
###
..#
###

input
3 4
output
####
...#
####

input
5 3
output
###
..#
###
#..
###

input
9 9
output
#########
........#
#########
#........
#########
........#
#########
#........
#########

Codeforces Round #290 (Div. 2)
Finished
Practice

→ Virtual participation
Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.
Start virtual contest

→ Practice
You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

→ Clone Contest to Mashup
You can clone this contest to a mashup.
Clone Contest

→ Submit?
Language: GNU G++17 7.3.0
Choose file: Choose File No file chosen
Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.
Submit

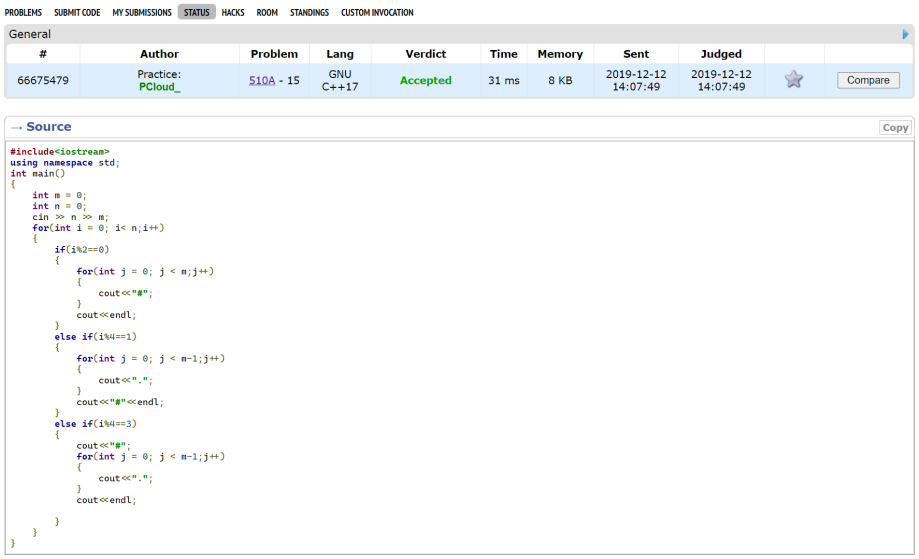
→ Last submissions

Submission	Time	Verdict
66675479	Dec/12/2019 14:07	Accepted
66675370	Dec/12/2019 14:05	Wrong answer on test 1
66675290	Dec/12/2019 14:03	Wrong answer on test 3

→ Problem tags
implementation *800
No tag edit access

The questions and the examples take up nearly all the spaces on the question page. There is no online editor or online runtime environment provided. Users are expected to write and test their code in their IDEs and only submit the solution for judging. Custom IDEs may be more powerful than a built-in one. My solution will provide an editor, it is much more convenient to use. Even if a user decides to use his environment, he can paste his code into the editor for submission. It sets the time and memory limits for users' submissions, if a piece of code takes too long to run, or takes up too much memory while running, it will be terminated and marked wrong.

Submission



PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged		
66675479	Practice: PCloud_	510A - 15	GNU C++17	Accepted	31 ms	8 KB	2019-12-12 14:07:49	2019-12-12 14:07:49		Compare

→ Source Copy

```
#include<iostream>
using namespace std;
int main()
{
    int m = 0;
    int n = 0;
    cin >> n >> m;
    for(int i = 0; i< n;i++)
    {
        if(i%2==0)
        {
            for(int j = 0; j < m;j++)
            {
                cout<<"#";
            }
            cout<<endl;
        }
        else if(i%4==1)
        {
            for(int j = 0; j < m-1;j++)
            {
                cout<<".";
            }
            cout<<"#"<<endl;
        }
        else if(i%4==3)
        {
            cout<<"#";
            for(int j = 0; j < m-1;j++)
            {
                cout<<".";
            }
            cout<<endl;

        }
    }
}
```

When the user submits the code, the code enters a queue waiting for judging, then the user can look up their result. Users can check their source code, performance stats, and more importantly, when they have not passed all test cases, they can see what they have got wrong. The judgment protocol provides detailed information about each test case, so users can debug easily.

→Judgement Protocol

Test: #1, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK
Input
aaabb
Output
6
Answer
6
Checker Log
ok 1 number(s): "6"

Test: #2, time: 15 ms., memory: 4 KB, exit code: 0, checker exit code: 0, verdict: OK
Input
usaco
Output
1
Answer
1
Checker Log
ok 1 number(s): "1"

Test: #3, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK
Input
lol
Output
2
Answer
2
Checker Log
ok 1 number(s): "2"

Test: #4, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 1, verdict: WRONG_ANSWER
Input
qdpinbmcrfwxpdbfgozvvocemjructoadewegtvbvbfwwrpgyeaxgddrwvlqnygwmwhmrhaizpyxvgaflrsvzhhzrouvpxrkxfza
Output
-763363328
Answer
37
Checker Log
wrong answer 1st numbers differ - expected: '37', found: '-763363328'

Analysis

Codeforces is optimized for coding competition, so it has a lightweight and complex interface for better performance. It is completely free to use. Users can create their questions but it is very complicated to do so. My solution needs to enable users without experience to create questions easily. There is no function for education - there is no way for a teacher to 'create a class' and monitor his students. Codeforces is an online platform, so it also works across all devices and requires an Internet connection. Users have to use their IDEs to write and debug their code. Codeforces sends all submissions to a central 'judging queue' for marking. My solution will mark all submissions locally, which makes judging a lot faster and save me from running a server. By limiting time and space allowance, Codeforces effectively prevents malicious code from running.

1.7 Features

A useful homepage interface with shortcuts to different functions in the software and other resources outside the software. This allows the user to get into practising faster and makes the software easy to use. Details about the design of the homepage will be discussed in the Design chapter.

A problem database with a graphical user interface. The GUI will have a search box and several drop-down menus for the user to input information to search for a problem. This provides the user with a simple way to find the problems they want, and it also ensures the users can manage and backup the data easily.

An interface for users to create new problems and share them with others. This interface will have multiple text fields for the user to input the descriptions to the problem, the expected input/output data. Then the problem is saved to the database and can be exported to a JSON file allowing the user to share it with others. Users can also create a 'list' of problems and export the entire list into a JSON file and share it with others. This enables teachers to create custom problems and share them with the students. It is a core feature that differentiates my solution from the existing ones.

An interface for teachers to create assignments. An assignment is a 'list' of problems with some extra data, such as the due date and total mark. It can be exported to a JSON file and share with students. The student's submission will be exported into a JSON file as well and can be sent back to the teacher. The solution will also integrate with the assignment function in Microsoft Teams for Education, which makes it even easier to do. The students' submissions will be automatically marked by the software and detailed data will be provided to the teacher. A simple data analysis interface will be provided to the teacher so they can have a brief look at the result. The teacher can also export the data into a CSV file so they can import it to their school system or analysis it with professional software. This automates the entire process from creating assignments, distribute assignments, collecting assignments, and marking

assignments. Teachers will have more time analysis the student's performance and provide corresponding help timely. It is a core feature that differentiates my solution from the existing ones.

An interface displaying the problem and the code editor. The solution provides a 'Run Code' button for the student to pre-run their code before submission, a 'Submit' button for the student to submit their code. This allows the students to read the question and write their code solution without switch between different windows. The 'Run Code' function also makes it easier to debug their code.

A playground with a code editor and runtime environment. This allows the software to be used in class teaching as well, the students can experiment with different algorithms and programming languages in the playground easily.

A settings page contains all the setting options for the software. Users can adjust settings such as their preferred programming language, syntax highlighting settings, colour themes, and so on. This allows the users to customize the software to fit their needs and allows users with different backgrounds to use it without issues.

1.8 Limitations

The software will be written in C# instead of web-based which means extra software needs to be downloaded by the user. I plan to use .NET 5 runtime and WinUI 3 library for my solution, so only the Windows 10 1809 or newer Windows operating systems will be supported. This should not cause many compatibility issues since most school computers are running the required version of the operating system. Downloading an extra software is inconvenient and may violate the IT security policy of some schools.

The judger can only accept code submission in limited programming languages and the user may require to configure their runtime environment. Creating a compiler for 'OCR Pseudocode Programming Language' is too complex for this project. I will attempt to allow the user to add their preferred programming language and write documentation for them to make the process easier.

Unlike LeetCode, there are no Discussion pages for users to discuss questions because it is a desktop program instead of a web one. But this is not a big problem, students and teachers should use an existing product such as Microsoft Teams which has very good support in sharing code snippets. It is unnecessary to rebuild the wheel.

Distribute the questions and assignments is still something inconvenient. Currently, distributing questions and assignments requires the teacher to first export the questions and assignments, then send them to the students through email or file-sharing platforms. When the students finish working, they need to send their results back through email or other apps. I have attempted to integrate the file-sharing function with the existing platform - the Microsoft

Teams Assignment function. ~~But unfortunately, the Graph API required for this operation is still in beta version, which means it can only be tested in the development environment and cannot be used in production. So for now, the users still have to use this inconvenient way to share questions and submissions. But in the future, the integration with some existing platforms may improve the experience.~~ (Update: the Microsoft Teams API is out of beta, now it is possible to integrate with it)

There are no good ways to maintain and distribute a large question database. Computer Science teachers are required to maintain a database for their students. But this is difficult work. Creating good test cases is much time consuming than writing a mark scheme, it is very likely for a wrong solution to pass the judging if the test cases are not good enough. It relies on the teacher who creates the questions to consider everything clearly to minimize its impact.

The judger can only simply compare the students' output with the expected output if there is a format error such as trailing space and extra newline in their output, which will not be considered as a mistake in a real exam, will be marked as a wrong answer by the judger. So students may need to spend extra time debugging their output format. It cannot judge "partially correct" answers as well. It does not care which line did the student get correct or wrong, if the final output doesn't match, the submission will be marked wrong.

1.9 Hardware and software requirements

Hardware and software requirements	Justification
Standard mouse, keyboard, and monitor.	Standard I/O devices are required for the user to interact with the software. Users need a mouse to navigate around different menus and pages, they need to use a keyboard to input their code solutions and use a monitor to get the output from the software.
Operating system: Windows 10 (1809 or later), Windows 11.	The software is designed with the WinUI 3 library and .Net 5 runtimes, which require such an operating system to run.
x86 64-bit CPU (Intel / AMD architecture) with 2 or more cores and 1 GHz or higher clock speed.	A modern CPU is required for the software. 1 core will be used to run the main program and at least 1 spare core is required for the judger to judge the submitted code. A clock speed higher than 1 GHz is required to ensure the software is running smoothly.
1GB free memory or more.	Around 512MB RAM is required to run the software, and another 512MB RAM is required for the judger to judge the submissions.
256MB free disk space or more.	256MB free disk space is required to store and run the program itself, the user may need extra disk space to store extra cache data and the database.
A modern dedicated or integrated graphics card.	The software has very little graphical demand, if the user's graphics card can run their operation system, it should be able to handle software as well.

1.10 Success criteria

Criteria	Justification
Users can use different links, menus, and buttons to navigate around the software easily.	This ensures the program is easy to use and allows the user to find the function they want to use quickly.
Users can use different drop-down menus and the search box to find a problem from the problem database.	This allows users to search for questions easily in the database.
Users can add new questions to the database.	This allows teachers to create new algorithm problems.
Correctly validate the new questions before adding them to the database.	Make sure correct data is input and prevent SQL injection.
Users can create lists of questions.	This allows teachers to organize problems better by creating lists to manage them.
Users can create assignments.	This allows teachers to create new assignments for their students.
Users can export/import questions, lists, and assignments from/to their problem database.	This allows the users to share questions and data with others easily.
Users can work on a problem and their submissions can be automatically marked by the judger.	This allows the users to practice and get feedback on the software.
Users can create submissions for assignments and export them into a file.	This allows the students to complete and hand in assignments easily.
Correctly access and interact with the Microsoft Teams API	Allowing teachers and students to manage assignments through Microsoft Teams.
The software can mark the assignments automatically.	This automates the marking process and reduces teachers' work.
The software can perform simple data analysis to the assignments data.	This provides teachers with an overview of student's performance on their assignments and allowing them to help their students better.
Users can export the assignment data to CSV files.	This allows the teachers to use advanced data analysis tools and import the data into their school system.
Users can use the playground to test any code.	This allows the users to experiment with new algorithms and programming languages and allows the software to be used in class teaching.

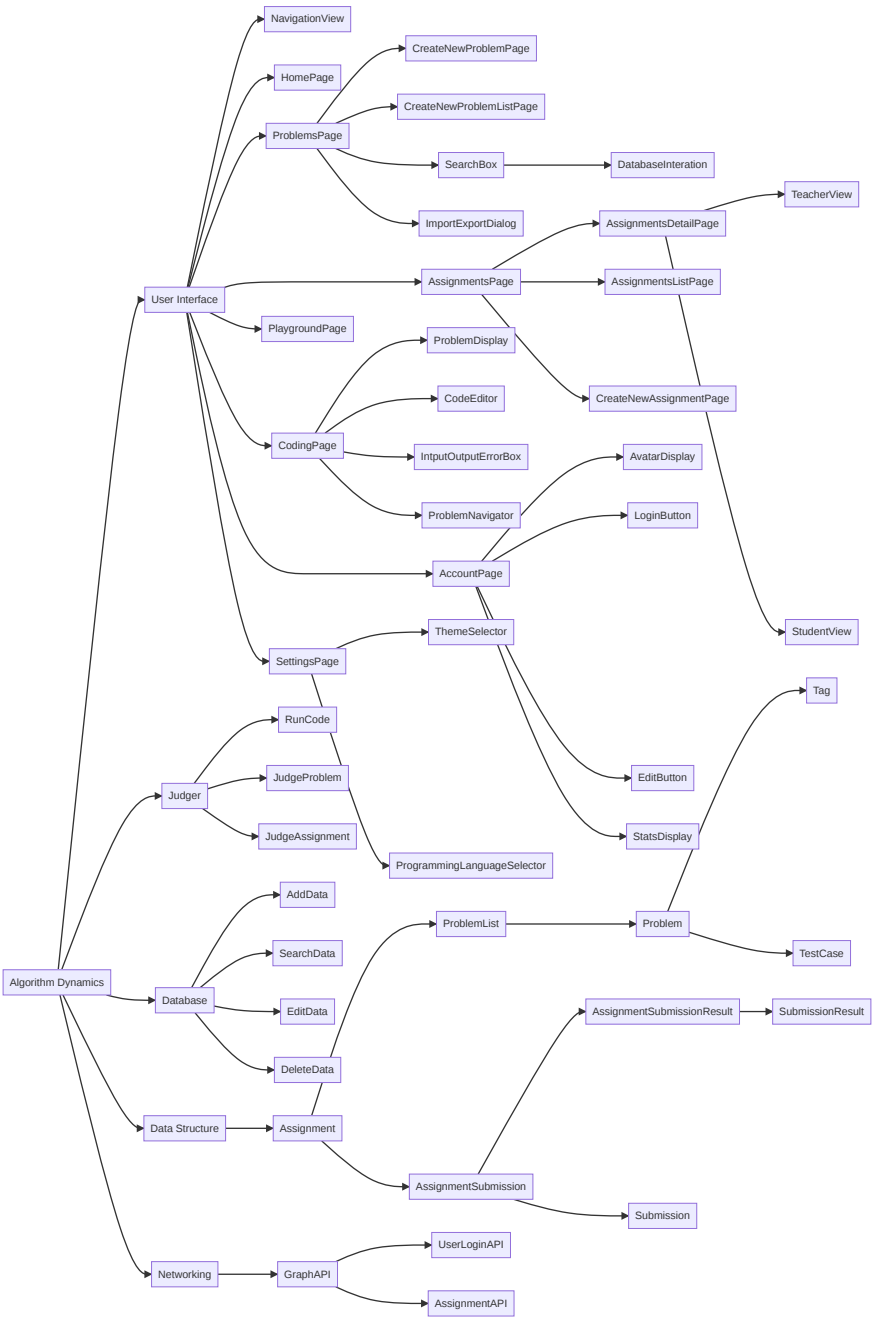
Users can customize the software.	This allows users to work in their favourite environment and makes the solution suitable to users with different backgrounds.
Split the core functions and class into a core library.	Allowing easier maintenance.
Using sensible variable names and add comments to each function.	Makes the code easy to understand and make maintenance easier.
Create CI/CD pipelines to build and deploy the application automatically.	It allows the software to be tested and deployed automatically so users always receive the latest features and security updates.
Unit tests with coverage higher than 90%.	It makes sure all code is well tested.

All the success criteria will first be tested through the unit tests created during the development. Then they will be tested and improved by me during the Alpha Testing stage. Finally, they will be tested by the stakeholders in the Beta Testing stage, and the evaluation will be based on their user experience and opinions.

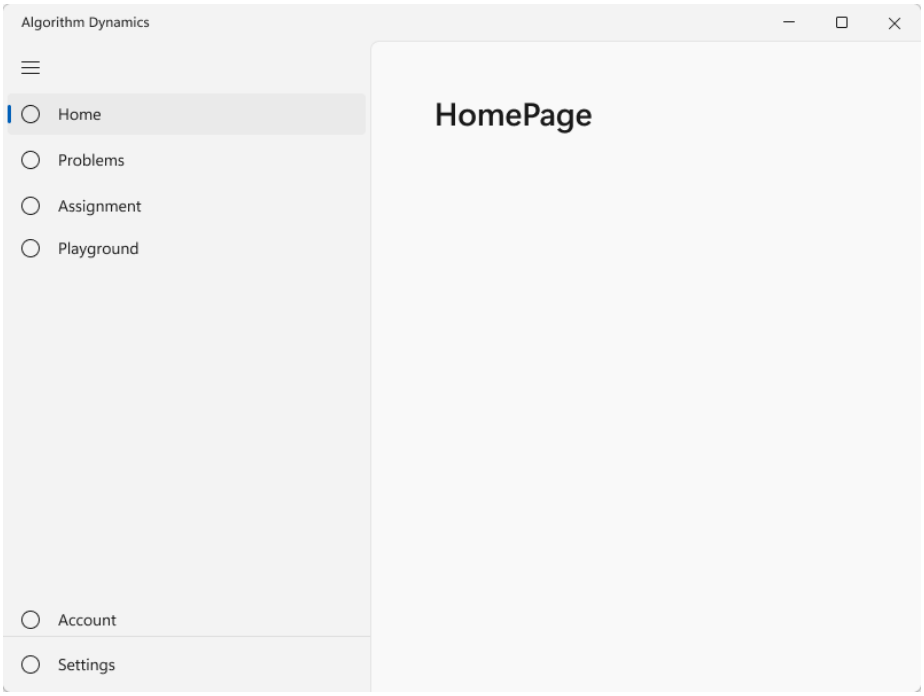
Chapter 2

Design

2.1 Decomposition


Algorithm Dynamics
User Interface
NavigationView
HomePage
ProblemsPage
CreateNewProblemPage
CreateNewProblemListPage
SearchBox
DatabaseInteration
ImportExportDialog
PlaygroundPage
CodingPage
ProblemDisplay
CodeEditor
IntputOutputErrorBox
ProblemNavigator
AccountPage
AvatarDisplay
LoginButton
SettingsPage
ThemeSelector
AssignmentsPage
AssignmentsDetailPage
TeacherView
AssignmentsListPage
CreateNewAssignmentPage
StudentView
Judger
RunCode
JudgeProblem
JudgeAssignment
ProgrammingLanguageSelector
EditButton
StatsDisplay
Tag
Database
AddData
SearchData
EditData
DeleteData
ProblemList
Problem
TestCase
Data Structure
Assignment
AssignmentSubmission
AssignmentSubmissionResult
SubmissionResult
Submission
Networking
GraphAPI
UserLoginAPI
AssignmentAPI

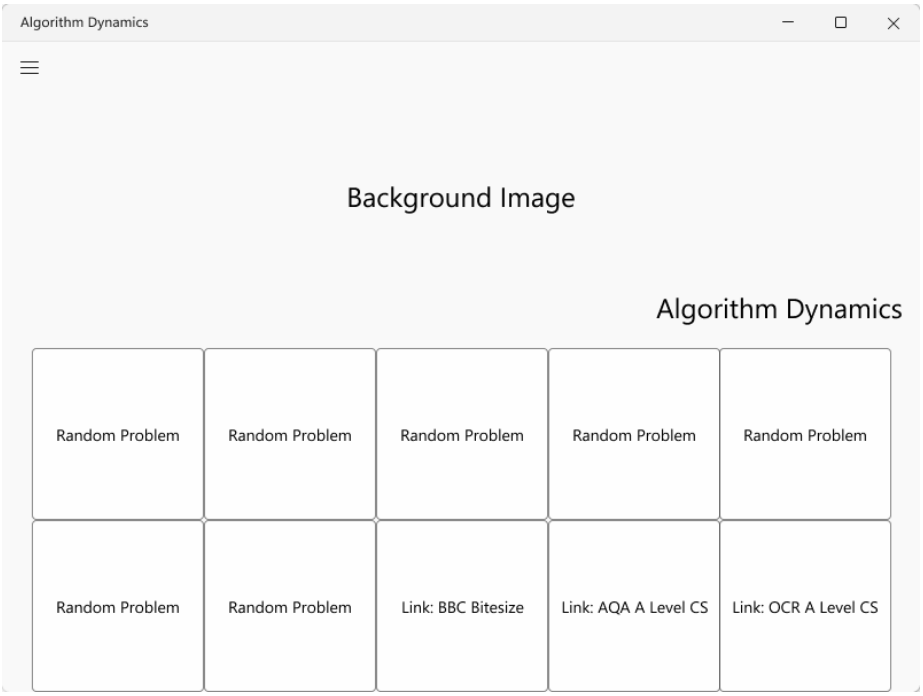
2.1.1 Navigation View



The Navigation View provides a global menu for the user to navigate between different pages in the software. There are six tabs in the navigation view, ”Home”, ”Problems”, ”Assignments”, ”Playground”, ”Account”, ”Settings”. By clicking on different tab, the main frame will display the corresponding page. The current selected tab will be highlighted.

Users will be able to know where they are and navigate to another page by one click, which makes the program easier to use. Since there is no input box in the navigation view, there is no need to validate user's input.

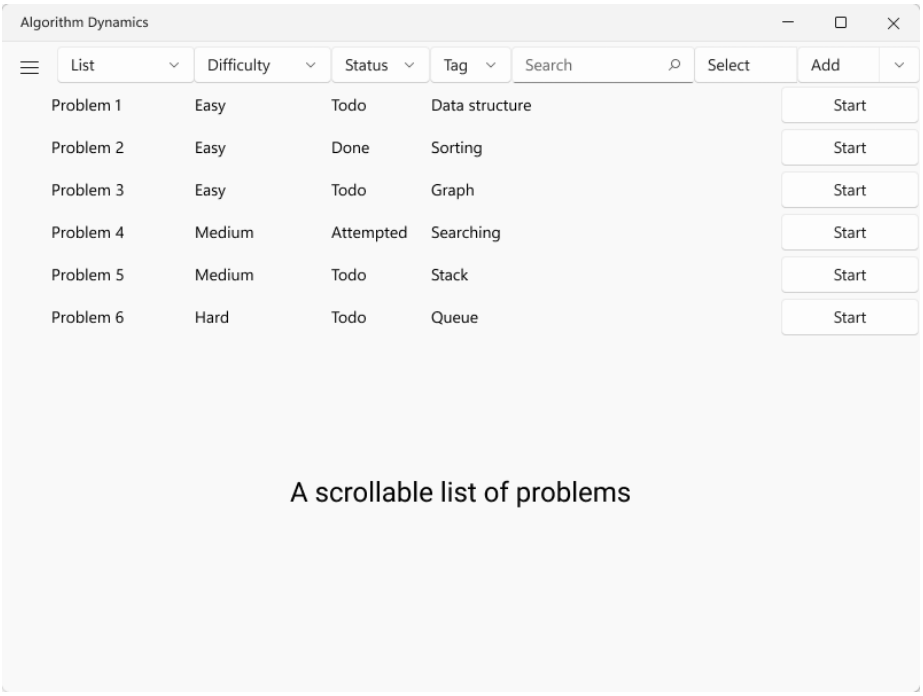
2.1.2 HomePage



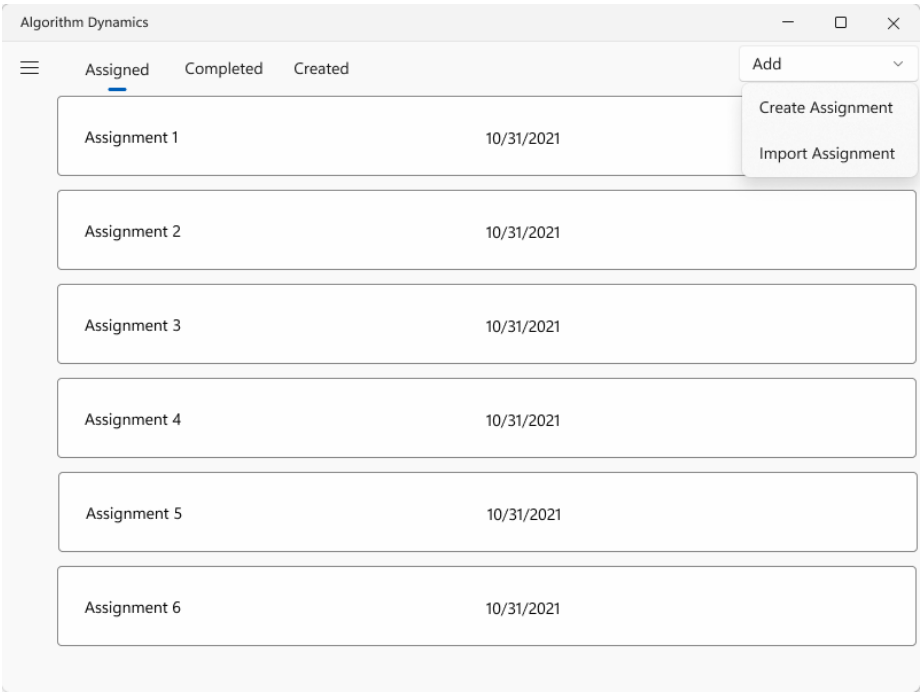
On the top of the home page, there will be a beautiful background image under the 'Algorithm Dynamics' title. This makes the interface looks pretty. Let users have a good mood every time they open this software.

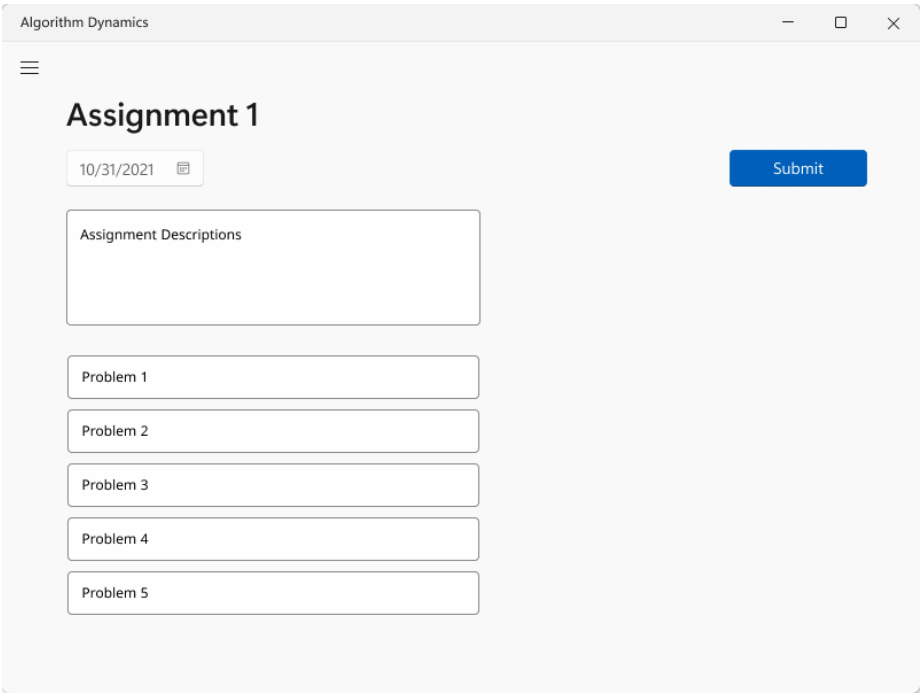
The bottom half of the home page will be filled with buttons link to different functions of the software. The 'Random' button starts a random problem for the user. The 'Import' button shows a window for the user to import problems, problem lists, or assignments. The 'Playground', 'Problems', 'Assignments' and 'Account' button links the user to the corresponding page. This groups all the useful functions of the software and putting them on the home page makes they easy accessible to the user, which makes the software easier to use.

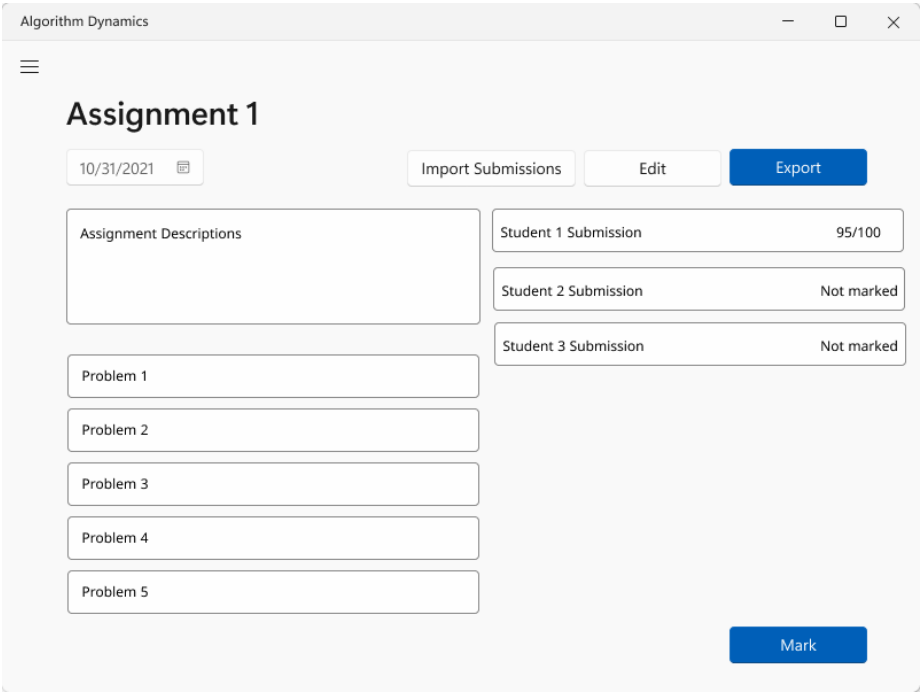
2.1.3 Problems Page



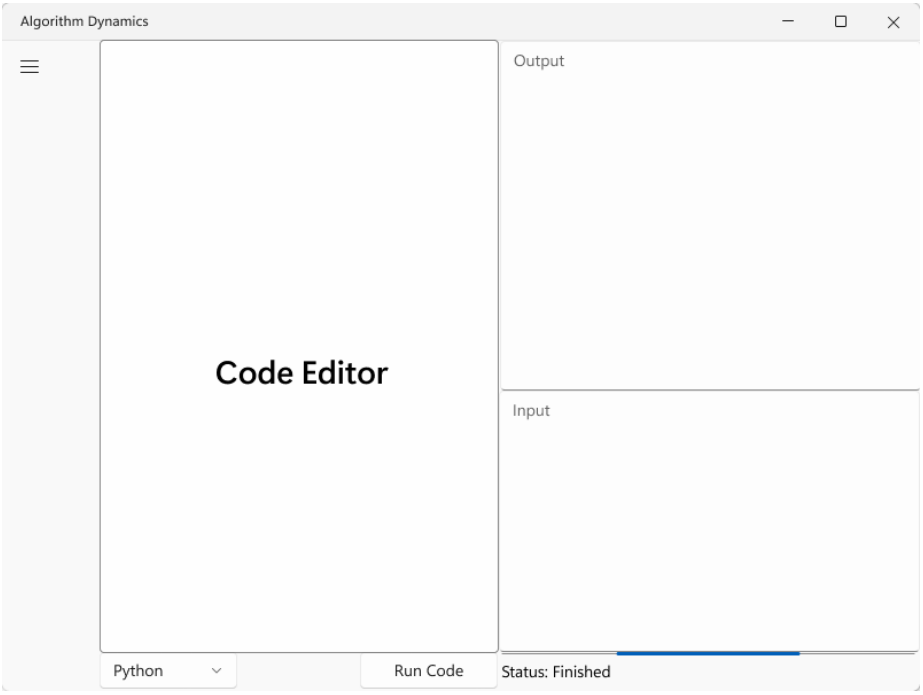
2.1.4 AssignmentsPage



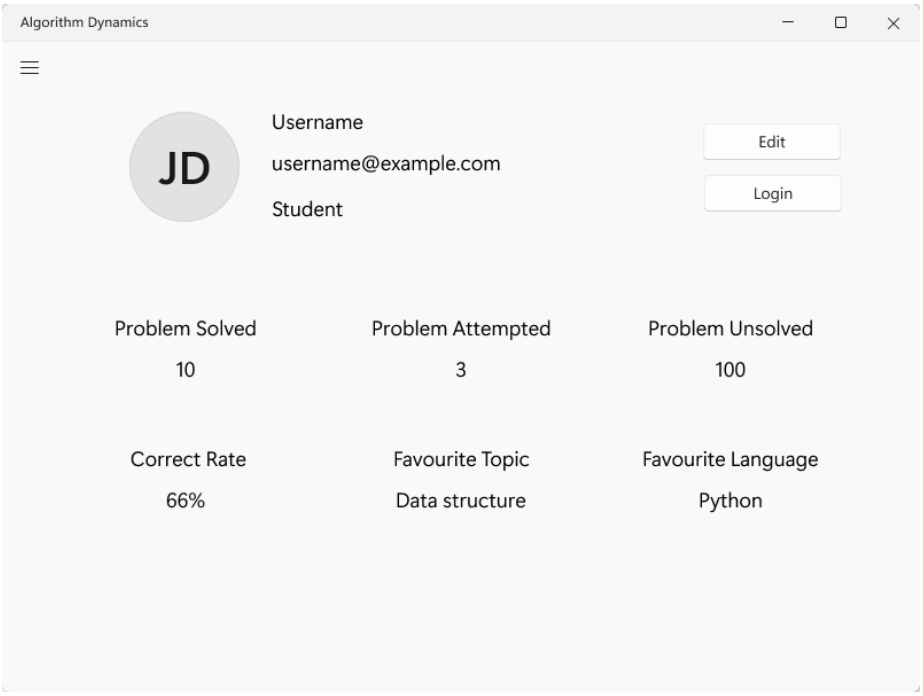




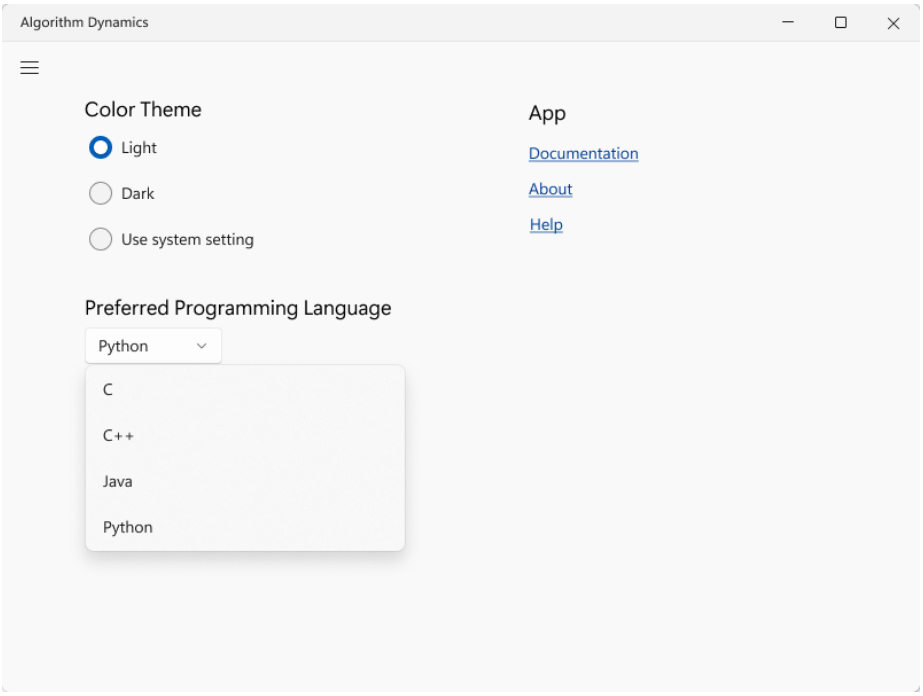
2.1.5 Playground



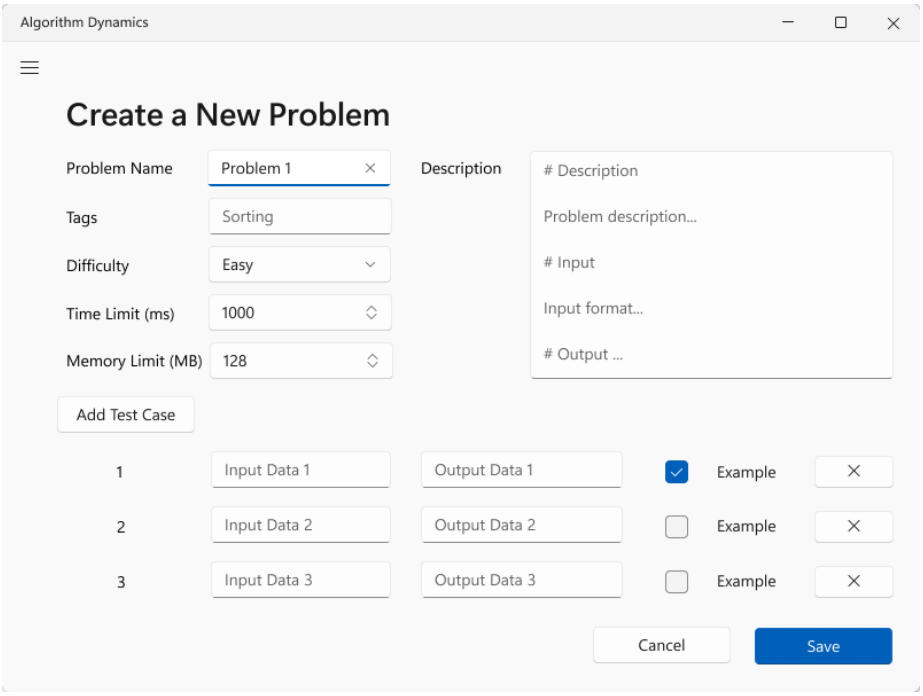
2.1.6 Account Page

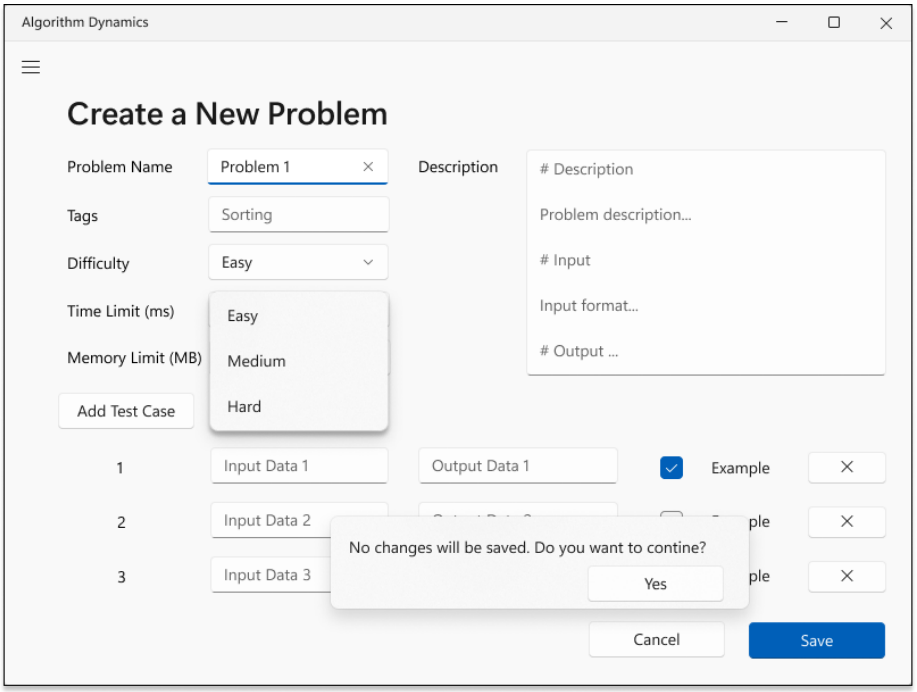


2.1.7 Settings Page

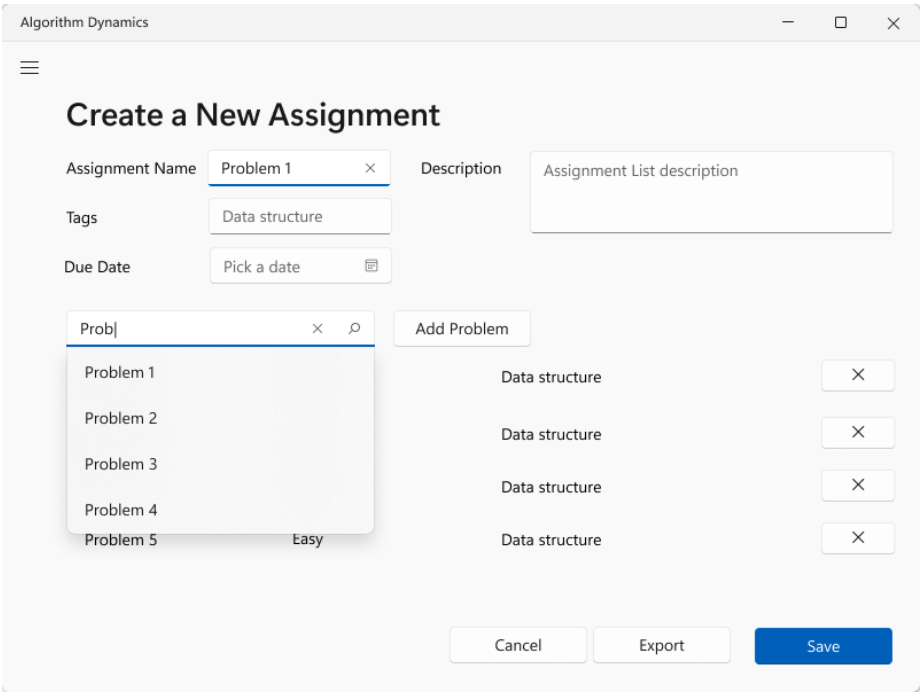


2.1.8 Create New Problem Page





2.1.9 Create Assignment Page



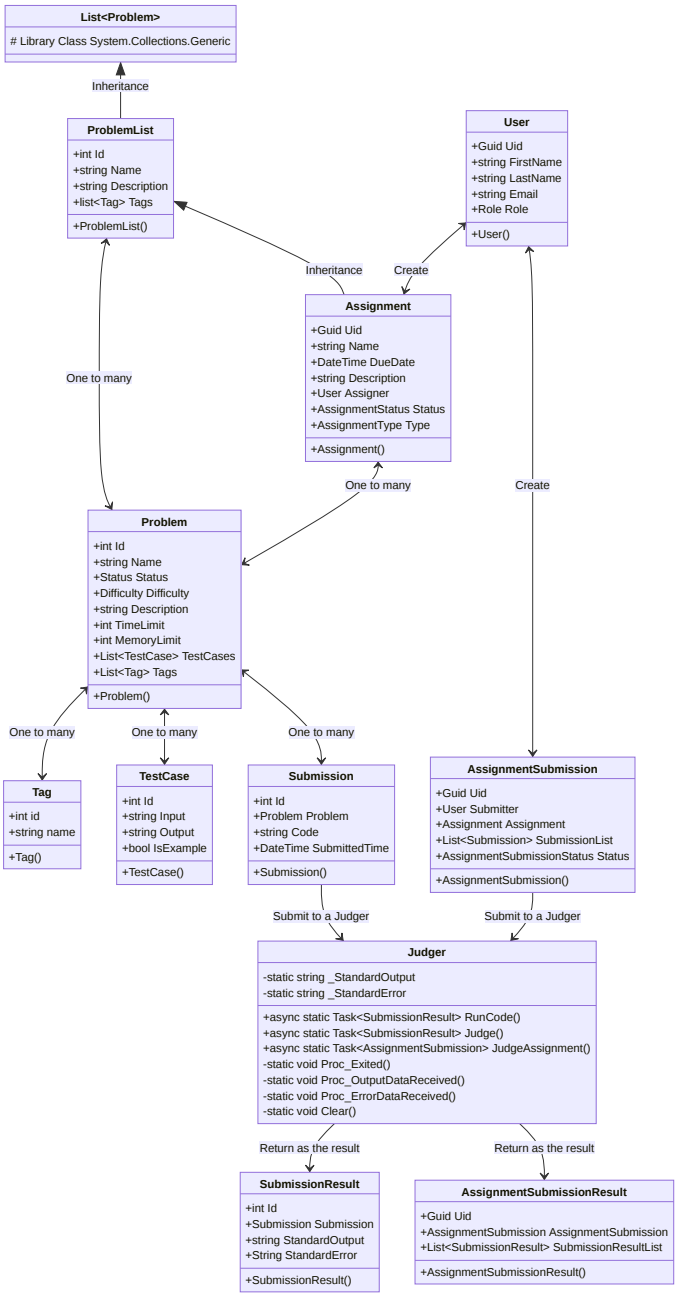
2.1.10 Judger Module

2.2 Algorithm design

2.3 Data structure design

2.3.1 Class design

I am taking an object-oriented approach to the design of the software. This is the class diagram for all classes.


List<Problem>
Library Class System.Collections.Generic
Inheritance
ProblemList
+int Id
+string Name
+string Description
+list<Tag> Tags
+ProblemList()
User
+Guid Uid
+string FirstName
+string LastName
+string Email
+Role Role
+User()
Inheritance
Create
Assignment
+Guid Uid
+string Name
+DateTime DueDate
+string Description
+User Assigner
+AssignmentStatus Status
+AssignmentType Type
+Assignment()
One to many
One to many
Create
Problem
+int Id
+string Name
+Status Status
+Difficulty Difficulty
+string Description
+int TimeLimit
+int MemoryLimit
+List<TestCase> TestCases
+List<Tag> Tags
+Problem()
One to many
One to many
One to many
Tag
+int id
+string name
+Tag()
TestCase
+int Id
+string Input
+string Output
+bool IsExample
+TestCase()
Submission
+int Id
+Problem Problem
+string Code
+DateTime SubmittedTime
+Submission()
AssignmentSubmission
+Guid Uid
+User Submitter
+Assignment Assignment
+List<Submission> SubmissionList
+AssignmentSubmissionStatus Status
+AssignmentSubmission()
Submit to a Judger
Submit to a Judger
Judger
-static string _StandardOutput
-static string _StandardError
+async static Task<SubmissionResult> RunCode()
+async static Task<SubmissionResult> Judge()
+async static Task<AssignmentSubmission> JudgeAssignment()
-static void Proc_Exited()
-static void Proc_OutputDataReceived()
-static void Proc_ErrorDataReceived()
-static void Clear()
Return as the result
Return as the result
SubmissionResult
+int Id
+Submission Submission
+string StandardOutput
+String StandardError
+SubmissionResult()
AssignmentSubmissionResult
+Guid Uid
+AssignmentSubmission AssignmentSubmission
+List<SubmissionResult> SubmissionResultList
+AssignmentSubmissionResult()

Tag

A Tag is a label used to categorize a Problem or a ProblemList.

Variable	Data type	Justification
Id	int	The Id is the local unique identifier for a Tag, it comes in handy when I need to store a Tag into the database.
Name	string	The Name of the Tag specifics the content of the Tag. I use string as the data type so a string of characters can be stored for its name.

TestCase

A TestCase is one set of test input and output, which will be used by the Judger to judge user's submission.

Variable	Data type	Justification
Id	int	The Id is the local unique identifier for a TestCase, it comes in handy when I need to store a TestCase into the database.
Input	string	The Input will be feed to the user's submission code by the Judger.
Output	string	The expected output of the input. The Judger will compare user's output with the Output to judge whether user's code is correct.
IsExample	bool	Define whether this TestCase is an example. An example TestCase will be displayed to the user for debugging, and distribute with an Assignment. A non-example TestCase will be used to judge the solution and will not be displayed to the user or distribute with an Assignment. A boolean value is suitable to store the two-state data.

Problem

The Problem is the core data object of this program. It is used to store and organize the data for each individual programming question.

Variable	Data type	Justification
`Id`	`int`	The `Id` is the local unique identifier for a `Problem`, it comes in handy when I need to store a `Problem` into the database.
`Name`	`string`	The `Name` is a string contains the name of a `Problem`.
`Status`	`enum Status`	`Status` is an enumeration type containing 3 possible status: `Todo`, `Solved` and `Attempted`. This is used to collect user statistics data. I choose to use a custom enumeration type instead of some magic numbers to make the code more readable and easier to maintain.
`Difficulty`	`enum Difficulty`	`Difficulty` is an enumeration type containing 3 possible difficulties: `Easy`, `Medium` and `Hard`. This provdes a way for the user to search and select problems by their difficulties.
`TimeLimit`	`int`	`TimeLimit` sets the max time allowed for the user code to run in millisecond. When the running time exceed the `TimeLimit`, the running code will be killed and a Time Limit Exceed error will be recorded. This prevents infinite loop from using up all computing resources and rejects inefficient algorithms.
`MemoryLimit`	`int`	`MemoryLimit` sets the max memory allowed for the user's code to consume in bytes. When the memory usage exceed the memory limit, the running code will be killed and a Memory Limit Exceed Error will be recorded. This prevents incorrect code from using up all memory space and rejects inefficient algorithms.
`TestCases`	`List<TestCase>`	`TestCases` is a list containing all `TestCase` for the problem. A list is more appropriate than an array because it allows new `TestCase` to be added or remove existing ones during runtime.
`Tags`	`List<Tags>`	`Tags` is a list containing all `Tags` for a `Problem`. Similarly, I use a list for `Tags` so it can be added or removed during runtime.

ProblemList

A ProblemList is a list of Problem, with a list of Tags so the user can share a list of problems easily. It is also the parent of Assignment and provides basic functions for it. The ProblemList is inheriate from the List<Problem> class, which is provded by the .NET 5 library. List<Problem> provides all basic functions for a list, such as add, remove, sort and find, so I don't need to reinvent the wheel. I choose a list instead of an array because the number of Problem inside the list will be changed during runtime, so a list is more appropriate for my use case.

Variable	Data type	Justification
Id	int	This Id is the local unique identifier for a ProblemList, it comes in handy when I need to store a ProblemList into the database.
Name	string	The name is a string to store the name of the problem list.
Description	string	The description is a string to store the description of a ProblemList.
Tags	List<Tag>	Tags stores a list of Tag for the ProblemList. Users can add new Tag to a ProblemList, which allows easier searching.

Assignment

An Assignment is a ProblemList with descriptions and a due date. Because the assignment needs to be distributed to students, it is a little more complicated. When an Assignment is distributed, a copy of that Assignment is created. All TestCases with IsExample set to false will be removed to prevent students from cheating. The Type of the Assignment will be set to Copy to indicate it is a distributed copy. The Judger will reject to judge a distributed Assignment and the AssignmentsPage will show the Assignment under the Assigned tab instead of the Created tab.

Variable	Data type	Justification
`Uid`	`Guid`	The `Assignment` will use a `Guid` value for its `Uid` instead of a normal `int` value to ensure the `Uid` is unique globally. So when the user import an `Assignment` into their database, the `Uid` will not conflict with any existing values, and it will not be changed (Unlike a `ProblemList`, for which will be assigned a new `Id` when importing). The `Uid` will be referenced by the `AssignmentSubmission` as well.
`Name`	`string`	The name is a string to store the name of the `Assignment`.
`DueDate`	`DateTime`	The `DueDate` stores the time for the due date.
`Description`	`string`	The description is a string to store the description of the `Assignment`.
`Tags`	`List<Tag>`	`Tags` stores a list of `Tag` for the problem list. Users can add new `Tag` to a `ProblemList`, which allows easier searching.

1. Tag: A tag is a label used to identify a problem.
2. Test case: A test case is one set of test input and output.
3. Problem: A problem is a programming question consists of problem description and test cases.
4. Problem list: A problem list is a list of problems. It allows to be searched, sorted, inserted new problems and deleted existing problems.
5. Assignment: An assignment is a problem list with due date and assignment descriptions.
6. Submission: A submission contains the submitted problem and the user code solution.
7. Submission result: A submission result contains the submission, the output of the submitted code and its result.
8. Judger: A judger takes in a submission, run the submitted code, create a submission result and return the result.

Chapter 3

Development

(TODO): Add development chapter

Chapter 4

Evaluation

(TODO): Add evaluation chapter