2017 – 2018  COMP 4801

AI Tutor for Programming Class (the education platform)

FYP Final Report

Group Members:

Chan Ka Yeung  3035181580
Kung Tsz Ho  3035176729
Lor Cheuk Hin  3035185108
Hui Pak Nam  3035185732

Supervisor: Dr. C.K. Chui

Date of submission: 15 - 04 - 2018
Content

Introduction ........................................................................................................................................ 8
   a) Background and Motivation ................................................................. 8
   b) Objectives ............................................................................................. 10
   c) Benefit of Work .................................................................................. 10
   d) Outline of Report .............................................................................. 11

Methodology ............................................................................................................................ 12
   a) Technology .......................................................................................... 12
      i) Python ............................................................................................. 12
      ii) Django .......................................................................................... 12
      iii) Django channels with Redis .......................................................... 13
   b) Skill Tree ............................................................................................. 13
      i) Skill Tree Structure ......................................................................... 13
      ii) Version Tree ................................................................................ 16
      iii) Profile Tree .................................................................................. 16
   c) System Overview ................................................................................ 18
   d) Database .............................................................................................. 19
   e) Analyser ............................................................................................. 21
   f) Clustering ............................................................................................. 22
      i) Procedures of Clustering ............................................................... 23
      ii) Data Cleaning ............................................................................... 24
      iii) Standardization .......................................................................... 24
      iv) Choice of Clustering Method (Preliminary Selection) ................. 25
      v) Choice of Clustering Method (Further Testing) ........................... 28
      vi) Clustering by Mean Shift .............................................................. 31
      vii) Analysis based on Clusters Formed .......................................... 34
      viii) Storing in Database .................................................................. 36
   g) AI Tutor (Advisor) .............................................................................. 37
   h) Peer Helping System ........................................................................... 38
   i) Exercise Suggestion Function ............................................................ 39
      i) Reenforcing Mode ......................................................................... 39
      ii) Learning New Skill Mode ............................................................. 40
      iii) Learning Specific Skill Mode ...................................................... 40
## List of Figures

Figure 1. Types of Education for Programming (from Stack Overflow Developer Survey 2017) ................................................................. 9
Figure 2. Skill Tree Structure .................................................................. 14
Figure 3. System Overview ................................................................. 18
Figure 4. Database Schema ..................................................................... 20
Figure 5. Comparison of Three Clustering Algorithms ......................... 29
Figure 6. Comparison of using Mean Shift and Affinity Propagation ........ 30
Figure 7. KDE Surface Example (from Atomic Object) ......................... 31
Figure 8. Registration Page ..................................................................... 41
Figure 9. Successful Registration .......................................................... 41
Figure 10. Homepage ............................................................................. 42
Figure 11. Room List ............................................................................ 42
Figure 12. Exercise Description Page ................................................. 43
Figure 13. Room Interface ...................................................................... 44
Figure 14. Result Display of ‘Save and Run’ ........................................ 45
Figure 15. Profile Tree ........................................................................... 45
Figure 16. Input Program Example for Analyser ..................................... 46
Figure 17. Output Result of Analyser ...................................................... 46
Figure 18. Result of Clustering ............................................................. 48
Figure 19. Programs in Cluster 2 ............................................................ 50
Figure 20. Advisor (Empty Program) ...................................................... 51
Figure 21. Advisor (Syntax Error) .......................................................... 52
Figure 22. Advisor (Logical Error) ........................................................ 52
Figure 23a. Screenshot of Exercise Suggestion Function ......................... 53
Figure 23b. Screenshot of Exercise Suggestion Function ......................... 54
Figure 24. Exercise List .......................................................................... 54
## List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>AST</td>
<td>Abstract Syntax Tree</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>KDE</td>
<td>Kernel Density Estimation</td>
</tr>
</tbody>
</table>
Abstract

More and more people start to learn programming at the present time. However, teaching resources may not be enough to handle a large number of students. Some of the students start to use online platform to do self-learning on programming. There is quite a lot of online platforms for learning programming but most of them are only suitable for students who already have some programming background but not new learners. Therefore, we would like to build an online platform that is also suitable for new learners to use. Our platform is called Code Nova.

Code Nova includes some special features, AI tutor, peer helping system and exercise suggestion function to assist users to learn programming. The AI tutor can give suggestions to students while they encountered difficulties in doing exercises. The peer helping system enables students to seek help from other capable students. When a student presses the help button, Code Nova matches suitable students to help him/her and then they can have discussion and edit the program synchronously in a chatroom. Users can also use the problem suggestion function to find a suitable exercise to start with. These features rely on data analysis with machine learning and AI methods. In order to do analysis, the data is needed to convert into a self-designed skill tree structure. Therefore, we need to design an analyser to perform this conversion.

Code Nova provides lots of assistance to users to help them to learn programming. It is very suitable for people who would like to self-learn programming. It can help relieving the problem of insufficient teaching resources on programming. Moreover, Code Nova can be used as a teaching tool in a programming course or class. It provides more practices on programming to students. Teachers use it as a teaching tool to support teaching and their pressure could be relieved.
Acknowledgement

We would like to acknowledge the supervisor and advisor of our final year project, Dr. C. K. Chui. He has given a lot of opinions and suggestions to us during our meeting and provided timely help when we have encountered difficulties during the development of this project.
Introduction

A brief introduction of our project, including background and motivation, objectives, benefit of work and outline of the report will be given in this section.

a) Background and Motivation

More and more people start to learn programming in the current society as they realize the importance of programming. According to a news article of The Seattle Times, the number of computer science students in US is nearly doubled in the recent few years [1]. Moreover, according to a research of Bauhinia Foundation Research Centre, more and more countries encourage students to learn programming and England and Slovakia have made coding compulsory for primary students [2]. It is believed that the number of students in programming is growing rapidly. However, the teaching resources may not be enough to handle such amount of students. There are insufficient programming teachers and some of the teachers are not very experienced in teaching programming. According to an article from Marketplace, some schools in Nashville find difficult to hire a coding teacher and some coding teachers possess very basic programming skills only [3]. These problems may lead to low teaching quality.
Therefore, some of the students start to do self-learning on programming. According to the Stack Overflow Developer Survey 2017, self-learning is the most common learning method in programming [4]. 90% of the 30354 respondents said that they have tried self-learning to learn programming (Figure 1)

![Figure 1. Types of Education for Programming (from Stack Overflow Developer Survey 2017)](image)

Some of students would use some online education platforms to self-learn programming. There are quite a lot of online education platforms such as Codecademy, CodeForces, LeetCode etc. However, these platforms may not be suitable for new learners to use as they do not provide much assistance to new learners. These platforms only provide exercises and materials but they cannot provide timely help to the users. Users who have problem may need to search for answer by themselves. New learners may know nothing about programming and it is hard for them to finish a programming exercise by themselves. More helps should be provided to them.
b) Objectives

The objective of our team is to develop an online programming education platform which can provide timely help to the users so that it is also suitable for new learners to use. The platform is called Code Nova. It is implemented by Python using Django as web framework. It provides basic functions such as registration, login, administrator, profile etc. with an attracting interface. Furthermore, the website also includes some special and new features like AI tutor, peer helping system and exercise suggestion function to assist new learners to learn programming. These features make use of some machine learning and AI techniques to do analysis on submitted programs from users. Clustering will be used in our platform. Before doing data analysis, some preprocessing is needed to be done on the programs. Therefore, an analyser used to preprocess the programs is needed in our project.

c) Benefit of Work

Code Nova provides learning materials to users. Users can read the materials and gain some basic knowledge on programming. After they have finished reading the materials, they can start to do some programming exercises. Users can also use the exercise suggestion function to find a suitable exercise to start with. If they encounter difficulties while they are doing exercises, the AI tutor in Code Nova can provide some hints or suggestions to them. If the users encountered a complicated problem, they can seek help from other capable users by using the peer helping system. With the assistance of these features, new learners can also enjoy the fun of learning programming.

Code Nova is a very good learning tool for people to self-learn programming. It provides more assistances than some other online platform on the Internet such that it is suitable for new learners to use. The AI tutor, the peer helping system and the exercise
suggestion function can partially replace the need of teachers. This can help relieving the problem of high demand but low supply of programming teachers.

Moreover, Code Nova can also be used as a teaching tool in a programming class or course. Code Nova provides more opportunities for students to practice programming. Also, the assistance provided by Code Nova can relieve the pressure of the teachers.

d) Outline of Report

The remainder of the report proceeds as follow. First, the methodology of Code Nova will be explained, including some technology used, system design and the working principles of the features. Second, the results of our project will be reported. Screenshots of the user interfaces will be shown in this section. Third, it will be the future work of our project. Lastly, the report will be closed by a conclusion.
Methodology

In this section, the methodology of our project, including system design and working principles of the major features will be introduced. My major contribution is on the clustering and the advisor function.

a) Technology

i) Python

Python is used for developing this project. It is a very popular programming language, especially in program development related to AI or machine learning. There are many useful libraries in Python, such as numpy, pandas, scikit-learn etc., which provide a lot of powerful functions for data processing and data analysis [5]. Python makes machine learning become much easier and more convenient. Therefore, Python is chosen for developing our project.

ii) Django

Code Nova is an online learning platform and web framework is needed to handle some basic functions like authentication. Django is chosen to be the web framework in this project as it is the most popular web framework for developing websites using Python. Moreover, Django also has the advantage that it prevents a lot of common security mistakes [6].
iii) Django channels with Redis

Code Nova is an online learning platform and thus it needs websockets to keep connections between clients and server. Also, websockets are needed to transfer information between clients and server. Redis is a popular tool used to manage the websockets in the backend system and it is convenient to use so it is chosen to be used in our project.

b) Skill Tree

Skill tree is a self-designed tree structure in Code Nova used to record some important and common programming skills such as if-else, loop, array, functions etc. Skill tree facilitates data analysis in Code Nova. In our project, there are two kinds of skill trees, version tree and profile tree. Version tree and profile tree have the same structure but they are used in different functions and generated by different methods.

i) Skill Tree Structure

The skill tree is a data structure used to record programming skills used in a program (version tree) or possessed by a user (profile tree). There are 144 nodes in the skill tree totally. The programming skills are categorized into different categories such as basic IO, condition, loop, class etc. The skill tree structure is shown in Figure 2. Nodes in the same indentation layer are in the same layer inside the skill tree.
Basic IO has two children, basic input and basic output, which are the most common and basic programming skills. Condition includes ‘if’ (further divided into ‘with else’ or ‘without else’) and switch. Loop includes single loop (including for loop and while loop) and nested loop (including for loop only, while loop only and mixed). Array is classified by whether it is a character array or non-character array and whether it is single dimension array or multi-dimension array. Function is classified by whether it is a function or a procedure and whether it contains recursion or not. Procedure means a function that has no return value. Class is classified by whether it has a constructor and whether it has inheritance property. Lambda means lambda function. Comprehension is divided into list comprehension, set comprehension and dictionary comprehension. Comprehension is a special and convenient way to define a data structure.

Data structure is also an important component in a program. In the skill tree, data structure is divided into list, set, tuple and dictionary. Moreover,
list is designed to contain some common library functions on list operation such as append and extend as list is the most frequently used data structure in Python. String (py-str) also contains some commonly used library function on string such as find, count, split etc.

Python also has a lot of built-in functions where some of them are very useful. Therefore, our skill tree also includes Python built-in functions such as len, range, eval etc. A point noteworthy is that some built-in functions are similar to some other skills in the skill tree and thus we have grouped them together. For example, the print function should be grouped with the basic output and the input function should be grouped with the basic input.

Module contains imported library function. In the current version of Code Nova, it only contains one library, collections. Some functions in the collections library are included such as deque, chain map etc.

Lastly, there are four special nodes, maxIfDepth, maxLoopDepth, maxArraySize and maxArrayDim. These four nodes have a different generation method. For other skills, they are generated by counting the frequency of use in the program. For these four nodes, they are unrelated to the frequency of use of programming skills.

maxIfDepth means maximum depth of ‘if’ structure. It records the depth of the deepest ‘if’ structure in the program. maxLoopDepth means maximum depth of loop structure which is similar to maxIfDepth. It records the depth of the deepest loop structure. maxArraySize means maximum array size. It records the number of elements in the largest array used in the program. maxArrayDim means maximum array dimensions. It records the dimensions of the highest dimension array in the program. These are the
four special skills in the skill trees and they can also help analysing and understanding the program structure of a program.

This is the designed skill tree structure. It is designed after several testing. The skill tree can help us represent a program or a profile of user in a simpler form and facilitate analysis process.

ii) Version Tree

A version tree records the programming skills used in a program submitted by a user. In Code Nova, the system needs to find out some common patterns among the programs submitted by users. However, it is quite difficult to compare different programs directly as everyone has different approaches and coding style towards a programming problem. Therefore, version tree is needed.

Every program is converted to a version tree by the analyser function before doing analysis. Then, all programs share a same data structure, version tree. The system can just analyse the version trees and find out some useful information. The version tree makes the comparison between programs becomes possible. Also, the version tree can be used as a data vector to put into the clustering algorithms. Version tree plays an important role in Code Nova.

iii) Profile Tree

A profile tree records the programming skills that a user possessed. Every user has a profile tree. After a user finished an exercise and submitted a correct solution, the profile tree will be updated.
The update rule of the profile tree is very simple. Initially, all nodes in the profile tree have value 0 as the user has not finished any exercise yet. After the user finishes an exercise, the system will convert the program into a version tree to see what programming skills are used in this program. For each programming skills used, the value of this skill in the profile tree will be increased by 1. For a skill in the profile tree, the higher the value of it means the more times the user used this skill (more skillful).

For the four special skills, the system will just record the highest value. For example, if maxIfDepth is 3 in the program submitted and it is 2 in the profile tree, then the maxIfDepth in the profile tree will be updated to 3. If the value in the profile tree is higher, no update is needed.
c) System Overview

Before introducing the major features of Code Nova, it is better to have an overview on the entire system. The system overview is shown in Figure 3.

![Figure 3. System Overview](image)

After a user logs in to Code Nova, he/she can choose a programming exercise from the exercise list or use the exercise suggestion function to choose an exercise. Then, he/she will enter a room to work on the exercise. Inside the room, the user can read materials and use the text editor to finish the program.

After the user finishes and submits the program, the system will save the program and try to run the program to see if there is any compilation error. If yes, the system will pass the error message to the AI tutor and the AI tutor will report the message to the user. If not, the system will test the program with some test cases. If the program
can pass all the test cases, this is the correct solution and the code will be stored in the correct solution database for further analysis using the analyser and the clustering function. The profile of the user will be updated.

If the program cannot pass all the test cases, the user needs to modify the program. The user can use the advisor function to get some hints. The system will compare the program of the user with the expected solution and try to find out differences between them. The differences may be the source of the errors. The detailed working principle of the advisor functions will be explained in later section.

If the advisor function is still not enough to help the user to get over the exercise, the student can use the peer helping system to seek help from other capable users. The system will match a suitable user to join the room. Then, these two users can discuss in the room by text messages and work together in the collaborative text editor. The working principles of the peer helping system will be explained in later section.

d) Database

Database is a necessary component in a project. Figure 4 shows the database schema of Code Nova. There are totally six kinds of entities in Code Nova. Each of them has several attributes and there are relationships between them. In the figure, ‘PK’ means primary key and ‘FK’ means foreign key. Each entity in the schema has primary key ‘id’ to uniquely identify an entity.
In the current version of Code Nova, there is only one type of user which is student. Each user has user name, password and email address as attributes. For student type user, each of them will have a profile to record information of the user.

A student can open a room for doing exercise. Each room is linked to one exercise only. The code writing in the room, participants of this room, chat history and whether this room is using the peer helping system (require_help) are stored. For an exercise, title of the exercise, problem description (material), test cases used to test the program, template code prefilled for user and the used skill set (common_skill) in this exercise are stored.

Also, an exercise will have several clusters. For each cluster, number of data, cluster center, necessary skills, redundant skills, characteristic skills and special skills (other_skill) will be recorded. The meaning of these skills and how the cluster is formed will be explained in the section of clustering function.

A version is a submission of program without syntax error (may not be a correct solution) for an exercise. It is linked to many entities. The database records which room, exercise and cluster do this version belongs to. Moreover, the code of this version, result
of testcases testing, version tree and whether this program passes all the testcases (overall_success) are recorded.

e) Analyser

Analyser is a function used to convert code into a version tree. As mentioned above, every program submitted by users will be converted into version tree before doing analysis. Analyser is the function responsible for this conversion. It can find out the programming skills used in a program and then record them in a version tree.

In order to convert a program into a version tree, the analyser first tries to compile the program to see whether the program contains syntax errors. If yes, the analyser will halt and display the error to the user. If not, the analyser converts the program into an abstract syntax tree (AST). The AST can show the program structure in a hierarchy structure.

By analysing the AST structure, the frequency of using different programming skills can be calculated. If for-loop is used for 3 times in the program, 3 for-loop can be found in the AST structure and thus we can know the frequency of using for-loop is 3. All leaf nodes (except special skills) are generated by this method. However, the non-leaf nodes are not generated by this way. They have summarizability property which means they are generated by summing up the leaf nodes value. For example, if the ‘for’ node is 5 and ‘while’ node is 3, then ‘single’ will be 8. If ‘single’ node is 8 and ‘nested’ node is 6, then the ‘loop’ node will be 14.

For the four special skills (maxIfDepth, maxLoopDepth, maxArraySize, maxArrayDim), the detection is a bit more complicated. For maxIfDepth and maxLoopDepth, the detection is done recursively. The AST structure has a hierarchy structure. It shows nested structure in the program. Recursively looping in the AST structure can find out the depth of ‘if’ and loop. The maximum value of ‘if’ depth and
loop depth will be recorded in the version tree. For maxArraySize and maxArrayDim, the system needs to keep track on the array structure in the entire program. Whenever the size or dimension of any of the array structures is changed, the change will be recorded. At last, the maximum value of array size and array dimension will be recorded in the version tree.

All information will be put into a JSON object and this JSON object is the version tree. JSON object is a very convenient data structure. If you want to retrieve a value from a JSON object, you just need to provide the corresponding key. The system can retrieve the frequency of using a particular programming skill by using the skill name. Moreover, the JSON object can be converted into dictionary object in Python very easily. The skill tree is stored into the database in JSON object and converted into dictionary to do analysis in Python.

There is a limitation for the analyser function. The analyser can only work on program without syntax error. The program must not contain syntax error. The reason is that the function used to generate AST requires the program to have no syntax error. If the program has syntax error, no AST can be generated and thus we cannot analyse it. This is the reason why we need to check whether the program has compilation error before using the analyser function.

f) Clustering

Clustering plays an important role in our platform. The objective of our platform is to assist the users, especially new learners, in learning programming by using the AI tutor. The AI tutor can provide some suggestions to the users if they have encountered some difficulties. However, it is not easy for the AI tutor to give appropriate advices to the user. Different people have different approaches toward a programming exercise and it is difficult for the AI tutor to classify different approaches. The AI tutor may give
some useless advices or even wrong suggestions to the users if it does not have enough information about the programming exercise. We hope that clustering technique can help us to analyse a programming exercise and provide more information for the AI tutor.

Clustering may be able to help us to classify the programs submitted by the users into different clusters and each cluster represents an approach for this programming exercise. The analyser captures the programming skills used in a program and converts each program into a version tree. Using the version trees as input data, clustering can help grouping those programs into some clusters. The programs inside same cluster should have some similar properties and common patterns. Every cluster is different and is representing an approach of this programming exercise. Based on these information, it is believed that the AI tutor can give more useful and accurate advices to the users.

i) Procedures of Clustering

After a user has submitted a correct solution for a programming exercise and this exercise has at least 50 correct submissions from the users, the system will analyse this exercise by clustering.

Firstly, the codes of this exercise are loaded from the database. Secondly, the codes are converted into version trees by the analyser and flattened into a data vector with 144 elements. Then, these data vectors will go through data cleaning and standardization. The data vectors will then be put together to form a data matrix. It is a $m \times n$ matrix where $m$ is the number of data vectors and $n$ is the dimension of the data vector after data preprocessing. After that, the data matrix is input to the clustering function provided by the scikit-learn library and the data vectors are divided into different clusters.
Finally, the system will find out some useful information based on the clusters formed. The system can find out the special patterns and characteristics of a cluster. The information will be stored into the database and will be used in the AI tutor (advisor) function which is the major feature of our project.

ii) Data Cleaning

After converting the programs into version trees, the skill tree will be flattened into a data vector with 144 elements. Then, the system will carry out a data cleaning process. The data cleaning process aims to remove some noise in the data. The skill tree has 144 skills. However, in a programming exercise, part of the skills may not be useful to solve this exercise. These skills may not be used by any user in this exercise. Therefore, these skills will have 0 value in all version trees of all users. If we include these skills in clustering, they may affect the result of clustering by a lot as they increase the elements in a data vector but having 0 in value in all data vectors. These skills are noise and should be removed from the data vector before doing clustering. If a skill is not used by any of the programs for this programming exercise, this skill will be removed from the data vector before clustering. The data cleaning process is very important for increasing the accuracy and efficiency of clustering.

iii) Standardization

After the data cleaning process, the data vectors will go through a standardization process which is also a kind of data preprocessing. Standardization does the function of feature scaling or data rescaling and it
is widely used in lots of machine learning algorithms [7]. In the standardization process, each feature in each data vector is first subtracted by mean of data and then rescaled to unit variance.

\[ x' = \frac{x - \bar{x}}{\sigma} \]

In the skill tree, not every node is having the same scale. The four special nodes and they are having different scale as other nodes. These four nodes are counted differently in the analyser. For other skills, they are counted by the frequency of used in the programs while these four are not. For example, maximum array size is related to the size of arrays created in the program. This is unrelated to frequency of using skill and it could be a very large number if the input is a large array. Therefore, standardization is needed to rescale the features.

In our platform, the standardization process is done by a function in the scikit-learn library which is called Standard Scaler (sklearn.preprocessing.StandardScaler). This function is convenient to use. Using the fit and transform method and using the data matrix as an input, the function will carry out standardization on all data vectors inside the data matrix.

iv) Choice of Clustering Method (Preliminary Selection)

There is a number of clustering methods provided in the scikit-learn library such as K-means, Mini Batch K-means, Affinity propagation, Mean Shift, DBSCAN, Spectral Clustering etc. Different methods have their own advantages and disadvantages. It is essential to find out an appropriate clustering method for this project.
After reading some documentations [8] [9] [10] about the properties of different clustering methods provided in the scikit-learn library and doing some preliminary test on them, it is believed that K-means, Mean Shift and Affinity Propagation are more suitable for our project.

K-means is easy to use and is suitable to use in many data set. Also, it performs quite well on the self-designed toy data set in the early stage of development. Therefore, it is thought that it is a candidate in our project.

K-means and Mini Batch K-means are similar clustering algorithms. Mini Batch K-means is more efficient than K-means in very large data set, like a data set larger than 100000. However, Mini Batch K-means may have slightly larger error than K-means as it uses a small batch of data to run K-means in each iteration [11]. In our platform, there may not be a large scale of data in the current development. Therefore, K-means is a better choice over Mini Batch K-means as it can provide a more accurate result.

The required input of the K-means algorithm is the expected number of clusters to be formed. During testing on this algorithm, we try different cluster number from 2 to 20 and see which cluster number gives the smallest error. The cluster giving the smallest error will be the best result. The error value can be retrieved from inertia_ which is one of the output value. inertia_ is the sum of all the squared distance between the data point and the cluster it belongs to [12].

Mean shift is also a popular algorithm in clustering. Different with K-means, mean shift does not need to input a cluster number while doing clustering. It can automatically divide the data into an appropriate number of clusters. Also, it will form uneven clusters [9] which is suitable in our project as we expected that the output clusters should be quite uneven in
size. Some approaches are more popular and some may be quite uncommon so the clusters should be different in size.

Mean shift has some optional parameters needed to be input but the most important one is bandwidth. Others are not very influential to the result or the default setting of the parameter is good enough. Bandwidth can be calculated by estimate_bandwidth, which is a function provided by the clustering library of scikit-learn. The estimate_bandwidth function requires the data matrix as an input. Furthermore, two more optional parameters quantile and n_samples may be needed.

Affinity propagation is another clustering algorithm. It has similar properties with mean shift. It does not need to input a cluster number and it will form uneven clusters [9]. These are good properties for our project.

There is a number of optional input parameters in affinity propagation and the most essential one is damping. Other parameters will not affect the result much or the default setting of the parameters is good enough. damping is a value between 0.5 and 1. Normally, if damping is smaller, the number of clusters generated will be fewer and vice versa.

For other clustering algorithms like spectral clustering and DBSCAN, they are not suitable to use in our project. Spectral clustering is suitable to use in a case that very few clusters are formed and the clusters are even [9]. These properties do not match with the use in our project. DBSCAN can form uneven clusters. However, DBSCAN is easy to classify some outliers as noise and leave them out of the clusters [10]. This is not a good property in our project. Some outliers in the data set may not be noise. They can also be some special and uncommon solutions. We would like to keep them in
the cluster. Therefore, spectral clustering and DBSCAN may not be a good choice.

v) Choice of Clustering Method (Further Testing)

Further testing and comparisons are needed to be done on K-means, mean shift and affinity propagation in order to make the final decision on the clustering method to be used in our project. We have found 382 programs on a sorting problem that submitted by participants in the CodeChef competition [13] to be a data set to test these three clustering algorithms.

The experiment procedures are as follow. The programs are first converted to version tree and flatten into data vectors. Second, all data vectors go through a data cleaning and standardization step. Third, the data vectors go through a principal component analysis (PCA) to reduce the dimension of the data vectors to 2-D. PCA is a common technique used to reduce the dimension of data. The reason of using PCA is for visualization of data so that we can compare different clustering methods by plotting graphs. The original data has a very high dimension and cannot be visualized by graph. In the real run of clustering, PCA will not be used as there will be some error during doing dimension reduction process. Finally, the data vectors will be put into different clustering methods to see the result.

The result of the testing is shown in Figure 5. The upper left graph showing the data after data cleaning, standardization and PCA. It shows the data distribution before doing clustering. The upper right graph is the result of using K-means. The bottom left graph is the result of using mean shift. The bottom right graph is the result of using affinity propagation.
(damping=0.95). For each of the methods, several times of testing are done in order to tune the input parameters to give a better result.

![Comparison of Three Clustering Algorithms](image)

**Figure 5. Comparison of Three Clustering Algorithms**

From the figure, the K-means method returns a result with 19 clusters. This is not reasonable and is not an expected result. Also, from the graph, it can be seen that some clusters are very close together but they are treated to be different clusters by the K-means method.

The reason of giving such a poor performance is that K-means needs to input an expected number of clusters and then the algorithm will base on the input value to generate clusters. However, in our platform, we do not know how many clusters should be returned. We could only run K-means by different input value (number of clusters) and compare the error of each run to find out which input value results a smallest error. However, generally, more the clusters will result a smaller overall error. When more clusters are allowed, the algorithm can fit the cluster centers into the data points well and result in a small error. Therefore, the K-means algorithm
returns a large number of clusters. This is one of the disadvantages of K-means algorithm. K-means is better to use in data that you already have an expected cluster number.

It seems that mean shift and affinity propagation give a better result. However, more testing is needed. In this experiment, PCA is used to help visualizing the data so that we can observe the differences of different clustering methods by plotting graphs. However, PCA leads to a larger error while reducing the dimensions of data. In the real run, PCA should not be used. Therefore, we need to try these two clustering methods without using PCA.

In this testing, we cannot compare the performances by plotting graphs as the data remains its original dimensions (144-D). We can only compare the performances on the number of clusters formed to see which one can give a more reasonable result.

Figure 6. Comparison of using Mean Shift and Affinity Propagation

Figure 6 shows the clustering result of using mean shift (left) and using affinity propagation (right). Mean shift gives 8 clusters and affinity propagation gives 32 clusters. Affinity propagation works much worse when the data does not go through the PCA process. The result given by affinity propagation has been tested for multiple times by adjusting the damping parameter. However, the number of clusters is still a large number.
It is not reasonable for a sorting problem to have such a lot of approaches. It seems that affinity propagation can only work well when PCA is done to the data. At last, we decided to use mean shift for doing clustering in our project.

vi) Clustering by Mean Shift

Mean shift is based on a concept called kernel density estimation (KDE) [14]. KDE is a method to find out the distribution of a set of data. KDE will generate a surface with some points being the peaks. The peaks are the center of the high-density areas. The shape of KDE surface depends on the dataset and the kernel bandwidth. For the same set of data, if the value of bandwidth is changed, KDE surface will also be changed. An example is shown in Figure 7.

![Figure 7. KDE Surface Example (from Atomic Object)](image)

Mean shift is a process that shifts the data points to the nearest peaks and the peaks will be the cluster centers at the end. The clusters formed depend on the KDE kernel and KDE kernel depends on kernel bandwidth. Therefore, bandwidth is an important value for mean shift. If bandwidth is small, the KDE surface will have more peaks and thus more clusters in
mean shift. If bandwidth is large, KDE surface has fewer peaks and thus less clusters in mean shift.

There is a number of input and output for the mean shift clustering. Firstly, it need to take the data matrix as an input. Secondly, the mean shift algorithm needs to input some parameters such as bandwidth, seeds, bin_seeding etc. All of these parameters are optional but they can affect the result of the algorithm. As mentioned above, bandwidth is the most important parameter. It can be generated by using the estimate_bandwidth function. seeds and bin_seeding are related to initialization of kernel. These parameters can be handled well by default function.

The estimate_bandwidth function also needs some input. It also takes the data matrix as an input. Moreover, it has two more optional but important parameters quantile (between 0 and 1) and n_samples. This function calculates the distance of every pair of data in the data matrix. Then, the distances are sorted and a quantile will be returned according to the quantile input [15]. If the input is 0.2, the 0.2 quantile is returned. The quantile is the most crucial factor and this will affect the clustering result by a lot. n_samples means how many samples in the data matrix are used to generate the bandwidth. It is common to set it to 100.

The mean shift algorithm will return two outputs, cluster_centers_ and labels_. cluster_centers_ is a list of the cluster center. Each cluster has a center. Each cluster center is a list of numbers (a vector) where the dimension of the cluster center is the same as that of the data vector. labels_ is list of number (a vector) where its dimension is same as the number of input data. It indicates which data belongs to which cluster. For example, if
the first element of labels_ is 0, that means the first data is belong to the cluster 0 (first cluster). labels_ clearly denotes the clustering result.

Adjusting the value of quantile becomes the most important task as the final result is deeply related to it. Besides using the sorting problem dataset, we have found 8 more datasets which are open source from the same webpage [13] and they will be used to tune the value of quantile. We need to find a suitable value that can fit these datasets well.

These datasets include The Lead Game Problem (229 data), Mahasena Problem (78 data), Carvans Problem (60 data), Chef and his Sequence Problem (59 data), Chef And Operators Problem (80 data), Three Way Communication Problem (95 data), Coin Flip Problem (97 data), Cops and the Thief Devu Problem (53 data). These problems are suitable for beginners and so they are suitable for our platform.

Testing is done on all of these datasets. For each dataset, we tried different value of quantile in estimate_bandwidth function and viewed the result of clustering, including the number of clusters formed, the necessary skill (skills that are used by most programs in the cluster), redundant skills (skills that are used by very few programs in the cluster) and the value of the four special skills for each cluster. Moreover, for each of the clusters, some programs are printed out. We checked whether all of these programs are using similar approaches or skills. Also, we checked whether these programs matched with the necessary skills, redundant skills and special skills of the cluster.

If the programs inside the same cluster contain similar skills and the skills are included in the necessary skills of the cluster. This result is desirable as the function can really group similar programs into a cluster.
This fulfills the objective of doing clustering. More details and information on clustering result will be given in the ‘Results’ section.

After doing the testing, it is found that a larger dataset has better clustering results when the quantile is large (~0.9) while a smaller dataset has better result when quantile is smaller (~0.4). This is an expected result. Larger quantile means larger bandwidth and larger bandwidth means fewer peaks in KDE surface and fewer clusters in mean shift. For a larger set of data, it is easier to have more peaks in KDE surface as the data is more likely to be scattered on a larger area. Therefore, it is better to set a larger bandwidth to tune down the number of peaks in KDE surface as well as the clusters formed in mean shift. However, we cannot set different parameters for different dataset manually. We need to have a fixed value for quantile. Finally, we decided to use 0.6 to be the quantile value. From the testing, quantile=0.6 can make a balance between large dataset and small dataset. It can return satisfactory result from large dataset as well as small dataset. It is believed that 0.6 is a suitable value for quantile after doing a lot of testing on these datasets.

vii) Analysis based on Clusters Formed

After the clusters are formed, more analysis is needed in order to summarize some useful information from the clusters. For each cluster, we need to find out the necessary skills, redundant skills, characteristic skills, mode of the four special skills and number of data inside the cluster.

Necessary skill means this skill is used by the majority of programs in this cluster. We set the threshold to be 65% after doing testing. That means a skill is a necessary skill if this skill is used in more than or equal to 65%
of the program in this cluster. It is believed that this skill is essential for this cluster. The necessary skill is ordered by decreasing order of the frequency of use of the skills. The most frequently used skill is put at the front of the skill list.

Redundant skill is the opposite of necessary skill. If a skill is merely used among the programs in a cluster, this skill may be a redundant skill or it may not be an important component for this cluster. We set the threshold to be 10% after several testing. If a skill is used in less than 10% of the programs of this cluster, this skill is a redundant skill. Moreover, the skills that are removed in the data cleaning process are also included in redundant skills as they are not used by any programs in this exercise.

Characteristic skill is generated based on necessary skills. The characteristic skills are the skills that are not the mutual skills of all clusters. Some of the skills may be necessary skills of all clusters. For example, basic input and basic output are very commonly used in almost all programming problems. Then, basic input and basic output are mutual skills of all clusters and they are not characteristic skills of any cluster. Simply speaking, characteristic skill is just the difference of necessary skill set and mutual skill set.

For the four special skills in the skill tree (maxIfDepth, maxLoopDepth, maxArraySize, maxArrayDim), they should be separated from the other skills and should not be included in necessary skill, redundant skill or characteristic skill. In the cluster analysis section, the mode of each of these four special skills in each cluster is needed to find out.

For example, in a cluster, there are 20 data having maxIfDepth 1, 35 data having maxIfDepth 2, 5 data having maxIfDepth 3, 1 data having
maxIfDepth 4. Then, the mode of maxIfDepth in this cluster is 2 as most programs in this cluster has maxIfDepth 2. We believe that the maxIfDepth of this cluster should be 2. We use the mode to represent the cluster. This method will be used for all four special skills.

The number of data inside the cluster is easy to find. It can be calculated by counting the number in the label_ variable returned by the clustering algorithm.

viii) Storing in Database

The final step of clustering is to store all the useful information generated into the database. Each programming exercise will store a number of elements. Firstly, the labels_ generated by the clustering algorithm will be stored. Secondly, the remaining skill set after data cleaning is also stored as we want to know which skills are participated in the clustering. The remaining skill set is also known as the used skill in this exercise. They are the skills appeared in at least one program in this exercise. Lastly, a list of cluster objects is stored.

Each cluster object contains some information about a cluster such as cluster center, number of data in this cluster, necessary skill, redundant skill, characteristic skill and mode of special skills. Mode of special skills are stored in a dictionary object. An example is {“name”: maxIfDepth, “mode”: 2}.

These information will be used in AI tutor or other functions.
g) AI Tutor (Advisor)

AI Tutor is a major feature in Code Nova. It can give advices to users when they encounter some difficulties during doing programming exercises. It is also called the advisor function. The advices are generated based on the information got in the clustering procedure. After a user asks for advices, the program of the user is first converted into a skill tree by the analyser. Then, the skill tree will be compared with each cluster. A set of advices will be generated for each cluster.

In the current version of advisor, there are three kinds of advices for each cluster. Firstly, it will provide advices on some skills that are lacked. A lacking skill means the skill is in the necessary skill set of the cluster but it is not in the version tree of the user. In this situation, this skill is said to be a lacking skill.

Secondly, some redundant skills will be provided as well. A redundant skill means the skill is in the redundant skill set as well as the version tree of the user. In this situation, it will be a redundant skill. The lacking skill and redundant skill are not applicable to the four special skills.

Thirdly, some advices will be given on the four special skills. For the four special skills, the mode value is stored for each skill for each cluster. If the value of these four skills are different with the mode value in the cluster, an advice will be given to the user. For example, if the mode value of maxIfDepth is 2 in the cluster but the value of maxIfDepth is 1 in the version tree of the user, the advisor will give an advice. The user may need to modify the ‘if’ structure to a nested one.

There is one special thing about the maxArraySize element. This element will not be used in all questions. In some question, the array size totally depends on the input value. For example, in a sorting problem, the array size is related to the input array of the testcases only. It may not be useful to discuss about the array size as it is variated
and we cannot tell what is the most suitable value for maximum array size. Another reason is that, when the analyser analyse a program, it will not put any input to the program. Therefore, the analyser cannot predict the array size when there is input. As a result, for these kinds of questions, maxArraySize will not be appeared in the advices.

Among all the clusters, two of them are more special. These two clusters will be labeled by an icon in the advisor function. The first one is the most popular cluster. It is the cluster having the largest number of data inside. Most of the programs of this exercise are inside this cluster. This cluster may represent the most popular approach to solve this question.

Another special cluster is the nearest cluster. This cluster represents an approach that is the most similar to the approach of the user. There are some steps to find out the nearest cluster. Firstly, the version tree undergone the same data cleaning process as that done in the clustering function. This version tree will then have the same dimension as all the cluster centers. Then, the Euclidean distance of the version tree and the center of each cluster is computed. Finally, the cluster having the smallest distance will be the nearest cluster.

h) Peer Helping System

Peer helping system allows users to find other users for help. In some situation, the AI tutor alone may not be enough to help the user to reach the correct solution as the problem may be too complicated. For this situation, the user (help seeker) can use the peer helping system by pressing the help button. The system will match a suitable user to be a helper.

There are some rules to match a helper. Firstly, the helper should have finished this exercise before. This user should have experience on this exercise so that we can ensure that the helper is able to finish this exercise and provide help to the help seeker.
Secondly, the system would try to find a helper that has the similar approach with the help seeker. To find the most similar helper, Euclidean distance is used again. The idea is similar with finding the nearest cluster. All programs in this exercise are converted into version trees and the Euclidean distance of these version trees and the version tree of the help seeker is calculated. The owner of the program with smallest distance will be the most appropriate helper.

After the system matched a helper, the helper is invited into the room of the help seeker. They can discuss in the room using the chat function. They can edit the program synchronously by using the synchronous editor.

i) Exercise Suggestion Function

The exercise suggestion function can give suggestion to users on choosing a suitable programming exercise. There is a number of programming exercises covered different coding topics in our platform and users may not know which exercise is more suitable for them to start with. For this case, user can use the exercise suggestion function. The function will analyse the profile of the user and compare the profile with the used skills set of the programming exercises to find a suitable exercise for him/her. Used skill means the remaining skill set after the data cleaning process in clustering function. There are three modes in this function.

i) Reenforcing Mode

If the user chooses ‘reenforcing’ mode, the system will suggest an exercise that the user has knowledge on all the used skill for this exercise. The means the skill set in the profile of the user is the same as the used skill set of this exercise. Users who want to practice more and reenforce the programming skills learnt currently can choose this mode.
However, if no exercise fulfills this requirement, we still need to suggest an exercise. We cannot return nothing to the user. The function will return an exercise which part of the used skills are possessed by the user. The user has some knowledge on this exercise and this is still suitable for the user to try on.

ii) Learning New Skill Mode

If the user chooses ‘learning new skill’ mode, the system finds an exercise that contains at least one skill that the user has not used before. That means the skill set of the used skills of this exercise is larger than the skill set of the profile of the user by at least one. The function will randomly pick and return an exercise from all the exercises that fulfill this requirement. User can try to use at least one new skill in the exercise.

iii) Learning Specific Skill Mode

‘learning specific skill’ mode is similar to ‘learning new skill’. The user can also try using at least one new skill in the exercise. The difference is that the user can choose the skill that he/she want to learn from the skill list in this mode. The system will find an exercise that the used skill set contains the chosen skill. However, the exercise can still contain other skills that are new to the user. The function will randomly pick and return an exercise from all the exercises that fulfill this requirement.
Results

All of the features and functions of Code Nova have been finished. In this section, the finished system will be reported and screenshot of the system will be shown.

a) User Interface

The user interface of different pages of Code Nova are shown in the following screenshots.

The following is the registration page of Code Nova (Figure 8). By using an email address and entering the password twice, a user can register an account in Code Nova. After a successful registration, a message will be displayed (Figure 9).

![Figure 8. Registration Page](image)

![Figure 9. Successful Registration](image)
The following is the homepage (after login) of Code Nova (Figure 10). On the top of the page, there is a bar with some tabs. The ‘index’ tab directs the user back to the homepage. The ‘Exercise’ tab directs user to the exercise page where user can use the exercise suggestion function or choose an exercise from list (shown in later section). The ‘List Room’ tab directs the user to a room list (Figure 11). In the room list, all rooms will be shown. For each room, it will display the owner of the room and the exercise name. A user can click ‘Enter’ to enter the room and work together with the room owner.

Figure 10. Homepage

Figure 11. Room List
If the user has chosen an exercise, he/she will be directed to the exercise description page (Figure 12). There is some information about the exercise, including description of task, input specification, output specification, sample input, sample output, variable constraints and the cluster list. The user can unfold a cluster to see necessary skill and characteristic skill of the cluster. In this page, users can get some information before doing exercise.

![Figure 12. Exercise Description Page](image-url)
Then, the user can enter the room to work on the program. There are three columns in the room interface (Figure 13). The left one displays the exercise information. The middle one is an editor where the user can write his/her program. The right one is the function panel and it contains some functions. The chat display area displays text message of this room, including welcome message, chatting text and messages from the AI tutor. The texting area is for user to type text message to communicate with other users inside the room. The ‘Save’ button is for the user to save the program. The ‘Save and Run’ button allows the user to compile the code. The system will use testcases to test the code and display the result in the chat display area and the user can open the result for more details (Figure 14). The ‘Search Helper’ button is for the user to use the peer helping system (more information will be given in later section). The ‘Ask Advice’ button is for the user to use the advisor function (details will be given in later section). The ‘Compile Results’ button allows the user to review the result of the last ‘save and run’.

Figure 13. Room Interface
From the upper right corner of the homepage, it is the logout button. On the left on the logout button, it is the email address of the user. By clicking on it, the user can view his/her profile (Figure 15). The profile is displayed in a tree form. By pointing on the nodes, users can view his/her value on that skill. By clicking on the nodes, the user can have a closer look of a branch.
b) Analyser

The analyser can take a program as an input and then output the skill tree structure of this program in JSON object (or Python dictionary structure). Figure 16 and 17 show an example of input and output of analyser.

```python
def quicksort(arr, start, stop):
    if stop > start:
        left = start
        right = stop
        pivot = arr[start]
        while left <= right:
            while arr[left] < pivot:
                left += 1
            while arr[right] > pivot:
                right -= 1
            if left <= right:
                arr[left], arr[right] = arr[right], arr[left]
                left += 1
                right -= 1
        quicksort(arr, start, right)
        quicksort(arr, left, stop)

arr = []
for i in range(0, 1):
    n = int(input())
    arr.append(n)
quicksort(arr, 0, len(arr)-1)
for elem in arr:
    print(elem)
```

![Figure 16. Input Program Example for Analyser](image1)

![Figure 17. Output Result of Analyser](image2)
We can compare the input program with the output result. There are 2 input() in the program so the value of basicIO_input is 2. There is 1 print() in the program so the value of basicIO_output is 1. Then, the analyser adds up these two value and records that basicIO is 3. There is a nested while loop in the program so the value of loop_nested_whileOnly is 1. Moreover, the maxLoopDepth is 2 due to this nested while loop. Similarly, the maxIfDepth is 2 due to the nested if structure. Furthermore, there is a self-defined recursive procedure in the program so function_recursion_procedure is 1. The result of the analyser matches with the program.
c) Clustering

As the dimension of the data is too high and PCA will cause errors, we cannot show the clustering result by plotting graph. In order to visualize the result, we can view the labels generated, the cluster number, the necessary skill and special skill of each cluster and takes some examples from the cluster to check the clustering result.

Using the dataset of sorting problem (382 data) as an example again to show the result of clustering (Figure 18).

![Figure 18. Result of Clustering](image-url)
The upper figure shows the labels returned by the mean shift algorithm. There are 8 clusters formed. Another figure shows the necessary skill list and special skill list of each cluster (ordered by the cluster index, the uppermost one is cluster 0). As we can see, necessary skill list of each cluster is different. Some of the cluster uses the sort function of list to do sorting while some of them do not. Some of them uses single loop while some of them uses nested loop. Some of them uses list comprehension, some of them uses eval function, some of them uses insert function, some of them uses append function. Each cluster has its own special features.

We can also check the relationship between the skill ‘if’ and the value of maxIfDepth. Only cluster 4 has ‘if’ in its necessary skill list so the value of maxIfDepth is 1 for cluster 4. For other clusters, the value of maxIfDepth is 0.

We can check the relationship between the skill loop and the value of maxLoopDepth too. Every cluster has loop inside the necessary skill. This is normal as loop is required when sorting a list of number. Some of them uses single loop and some of them uses nested loop. For those who use single loop only, maxLoopDepth is 1. For those who use nested loop, maxLoopDepth is 2. This is correct.
We can try to print the programs from a cluster to further check the result (Figure 19).

11 programs are classified into cluster 2. All 11 programs are shown in Figure 15. From the necessary skill of cluster 2, we found that the most special things about cluster 2 is that it has nested loop and it does not use the sort function. From these 11 programs, all of them do not contain sort function and all of them have used nested loop. This is correct.

Although differences between clusters are not very significant yet and there are still some errors, the result is still generally satisfactory. Some meaningful information can be found from the result.
d) AI tutor (Advisor)

A user can click the ‘Ask Advice’ button to get some advices from the AI tutor. A brief advice will be displayed in the chat display area in the room. The user can open the advice for more details. There is a pie chart showing the data distribution in different clusters. The light bulb icon means this is the nearest cluster. The icon with five people means this is the most popular cluster. The user can click on each cluster to see more information. Currently, the user has written nothing in the editor. The advisor gives some advices on what skills are needed, what skills are not needed and what is the expected loop depth etc. This gives a framework to the user. The user knows what should and what should not be included in the program. It is helpful to the user.

Figure 20. Advisor (Empty Program)
After the user have written something for the program, the advisor will give different advices. If the program contains syntax error, the AI tutor will report the error in the chat display area (Figure 21).

![Figure 21. Advisor (Syntax Error)](image)

If the program does not contain syntax error but it cannot pass all the testcases, the AI tutor will also give some advices to help the user to modify the program. For example, the user forgets to use the sort function in the program so the program cannot pass all the testcases. The AI tutor tells the user that sort function is lacked (Figure 22).

![Figure 22. Advisor (Logical Error)](image)

In both the most popular cluster and nearest cluster, the AI tutor tells the user to add sort function into the program. Although there may be some useless advices, such as advising the user to use for loop, the AI tutor still points out the problem of lacking sort function in the program.
e) Peer Helping System

After clicking the ‘Search Helper’ button, the system will try to find a helper and send a help request to the helper to invite him/her into the room. If the helper accepts the invitation, he/she will enter the room. The helper and the help seeker can use the chat function to chat. They can discuss about the problem. Moreover, the editor at the middle column of the room is a synchronous editor and it allows synchronous editing by multiple users. The helper and the help seeker can write program simultaneously in the editor.

f) Exercise Suggestion Function

After entering the exercise list page, three columns will be shown on the screen (Figure 23a). Each column represents a mode in the exercise suggestion function. User can click the area to choose a mode. Then, a question will be suggested.

Figure 23a. Screenshot of Exercise Suggestion Function
After clicking one of the columns, a pop-up window will be displayed to suggest the user to try an exercise (Figure 23b). If the user clicks ‘Confirm’, he/she will be directed to the exercise description page.

![Figure 23b. Screenshot of Exercise Suggestion Function](image)

If the user does not want to use the exercise suggestion function, he/she can also choose an exercise from the exercise list (Figure 24).

![Figure 24. Exercise List](image)
Future Work

Code Nova still have some limitations and shortcomings that needed to be solved and improved. This will be the future work of our project.

Firstly, Code Nova supports Python only in the current version. The system provides learning materials and programming exercises in Python only. Users who want to learn other popular programming languages such as C++, Java are not suitable to use the current version Code Nova. This is one of the shortcomings of our project. In the future development of Code Nova, we will extend our platform to support other popular programming languages so that users can use Code Nova to learn other languages. In our current development, most of the features can be extended to be used for other programming languages. We can modify the skill tree and the analyser. Then, the rest of the system can be adapted for other programming languages.

Secondly, the analyser cannot work on a program contains syntax error. No version tree can be generated for a program contains syntax error. Therefore, the advisor function cannot be used until the syntax error is solved. Although we have reported the syntax error to the user for them to solve the syntax error before using the advisor function, it is still better if the user can use the advisor function when his/her programs still contains some syntax errors. We could solve this problem in the future.

Thirdly, the current version of the AI tutor still has room to improve. The AI tutor only provides advices on lacking skills, redundant skills and the four special skills. It may not be enough to support the users for some complicated problem. Therefore, we would like to optimize the AI tutor to give more useful and accurate advices. Suggesting a skill is lacking may not be enough for the user. If the AI tutor can tell the user what code should be added to modify the program, it will be much more helpful.
In order to improve the functionality of the advisor, it is better to work on the clustering function. If the clustering function can return a more accurate result, the advisor can give better advices as the advices are given based on the clustering result. There is one way possible to improve the clustering result. We can add weighting to more important skills in an exercise. For example, if this problem is about sorting, then sort function and loop may be more important skills. Then, the weighting of these two skills will be higher. However, this will be a very difficult task as different exercises have different important skills. It is difficult to find out what skills are more important in different exercises.
**Conclusion**

Code Nova can provide lots of assistances to people who want to self-learn programming. Also, it can be used as a teaching tool in programming courses. It can help to relieve the pressure of teachers.

In Code Nova, there are some special features provided to help the users when they encounter problems. The AI tutor provides some hints and advices to help users to modify the program. The peer helping system allows users to seek help from other more capable users. The helper and the help seeker can discuss and write programs in the room. The exercise suggestion function can help user to pick an exercise to start with. These features are based on clustering and analysing the correct solutions collected for users. Mean shift algorithm is decided to be used for clustering after several testing and consideration. In order to do clustering on programs, we need to use the analyser function to convert the programs into version trees where version tree is a self-designed tree structure in Code Nova.

There are still some limitations for Code Nova. We hope that these problems can be solved in future development so that Code Nova can provide better help to users.
References


