Definition of data requirements (Design data dictionary – from the viewpoint of programmer)

* Description of record structure:

Long term data: The times achieved for timed runs of algorithms, forenames of the students who achieve them and the generated rank number will be stored in binary random files. An example of how data will be stored in the binary random file is like this:

1

John

00:00:00

Where the rank is stored as an Integer, the name is stored as String and the time achieved is stored as DateTime.

This example will then be displayed in the times achieved section for each timed run of an algorithm and the teacher’s section like this:

Rank Name Time

1 John 00:00:00

However the forename will be set to a maximum of 50 characters for each one saved, so that users don’t have too many problems with errors when they enter their forename; meaning that there is less for them to remember. This is due to the fact that the width and length of binary random files must be set for data to be stored in them.

The password for the teacher’s section will be stored in the coding and as long as the password entered in the textbox matches this then the user will be allowed access to the teacher’s section; to make changes to the data stored.

Short term data:

I will store the arc weights of the network in a 2D array, the weights will be decided by a random number generator. This is only to be done while the program is debugging and will therefore not be stored for the long term. This will make it easier to construct the table for Prim’s algorithm tabular form and it will also make it easier for my program to construct the network. Also all of the vertices of the network will be assigned a letter from A-G where only a maximum of one direct route will exist between each vertex. However there will be no loops in my network, therefore e.g.: A won’t have a direct connection to A. Therefore the array will be an 8x8 array because there are 8 destinations and only a maximum of one route between each vertex. Also the workings produced from the run or timed run of an algorithm will not be kept for the long term as they are not needed, however they will need to be stored during the run of an algorithm so that then the program can do the next stage.

* Validation required:

Long term data: When the student enters their forename they will have to enter one as string which contains no spaces. If they enter a forename that is not as string, contains spaces, contains no characters or contains more than 50 characters then the program will flag an error message (once the user has pressed the “OK” button); that requests the user to enter the forename again. It will also tell them what they have done wrong so that they know what to change for it to be accepted. This will be done by using a message box which will appear and say “This is an invalid forename because it either contains an invalid character such as a space or a number, it contains no characters or it contains over 50 characters. Can you please enter your forename again?” Also in the message box it will display a button that says “OK” and the user must press this to continue to enter their forename as a forename must be entered. Once this forename has been entered, the user has pressed “OK” and the forename has been accepted a different message box will appear which says “Your forename and time has been saved to the leader board” where the user will have to press an “OK” button on the message box to go back to the main menu. The time where the timer was stopped will be shown on the screen to the user throughout this process. However when the program produces an error message because the name entered is too long it will only happen if the name is ridiculously long because the limit has been made high to prevent this.

When the timer reaches 59:59:99 the program will stop and ask the user for their forename and after this has been accepted it will save a time of 59:59:99 as their time achieved. This is because in the exam students will not be allowed to take more than 30 minutes on these types of questions, however I have made it so that the student has one hour because they need to learn how to perform the algorithms and therefore may need more time than that is allowed to do this.

There is no limit to the number denoting the rank (which is generated by the bubble sort), however my program must be able to cope with at least 100 data items and store them in order.

For the password that is entered to be valid it must match the password given in the coding, however when the password is entered into the textbox by the user it will appear as black dots so that it can’t be seen by others. Once they have entered the password into the textbox and they have pressed the “Sign in” button the user will gain access to the teacher’s page; if they entered the correct password. If the user enters an incorrect password a message box will appear and it will say “The password you entered was incorrect. Please enter a different password to gain access to the teacher page.” Also the user will have to press the “OK” button to continue entering the password where upon doing this the contents of the textbox will be deleted; so that they can enter another password.

Short term data:

The arc weights must be in the range of values of 0<arc weights<=20 and they must be integers. However this will be implemented using my program so that the random numbers are multiplied by a constant so that what ever comes out is a number in this interval. Also the arc weight will denote the distance between a vertex and another vertex. Also the same arc weight can be used to denote more than one arc. The network will be set meaning that the same vertices will have the same letters for each run or timed run, however the arcs connecting them will have different weights.

* File organisation and processing:

Long term data: The files containing the saved forenames, times and ranks for each timed run of an algorithm will be split into four different binary random files, so that they can be easily updated and easily coded for. One binary random file will contain the times achieved for timed runs of Kruskal’s algorithm, another for Prim’s algorithm network form, another for Prim’s algorithm tabular form and the last for Dijkstra’s algorithm. These will be opened in list boxes in the teacher’s page of my program, therefore only teachers will be allowed to update the data stored in each file.

Also when the user chooses the times achieved option on the main menu for an algorithm the program will access the corresponding file and display the contents in a list box which the user will be able to scroll up or down, or left or right, so that they can see all of the times achieved. This will also be done on the teacher’s page, however in the teacher’s page teachers will be able to update and delete the data in the list boxes and therefore in the files themselves; once they have selected one of the options on the teacher’s menu.

The teacher’s section password will be included in the code so that it is just a case of writing code that accepts the password if the text entered in the textbox equals that in the code.

Short term data:

The arc weights and vertex letters will be contained in a 2D array during debugging and then the arc weights will be deleted once finished. The program will read this array (which has the arc weights contained) line by line.