**Building Code Using Arrays, GitHub and Teamwork**

Riley McCallay

In this report I will be discussing a project in which I was required to work in a team of three other students to collaborate and take the lead in writing individual codes, with each team member writing segments of code for each other as well. In order to accomplish this, we utilized GitHub and shared a repository that we all used together. In my own code, I created a program that obtains two user input arrays and then merges them together.

To go into further detail on what my code actually does, it prompts the user to define the size of two arrays, and then prompts the user to input values within the two arrays. I used custom methods in order to sort the input values in ascending order, to be printed back sorted. Once this is done, I used another custom method to combine the two arrays into a single array. Finally, the merged array is printed back, with all the values of the first two arrays sorted in ascending order. The code is as follows, with comments to explain the functions of each piece.

**import** java.util.\*; //utilities

**public** **class** SortandMerge { //class

**int**[] a = **null**;

**int**[] b = **null**;

**public** **static** **void** main(String[] args) { //main method

Scanner input = **new** Scanner(System.***in***); //allow input

//begin by calling the input and sorting method for the first array

*contributionFromTyler*(); // add print line to proceed

**int**[] a = *sort*(); // assing int a to the first input array

*next*(); // proceed to next array input

//call the method for second array input

**int**[] b = *sort*(); // assign int b to the second input array

*ContributionFromConner*(); // add print line to proceed

//move on to the merger

**int**[] c = *merge*(a, b); // declare a merger variable

System.***out***.println("The merged lists are now: ");

System.***out***.println(Arrays.*toString*(c));; // call the merge method

}

**private** **static** **void** next(){

System.***out***.println("");

System.***out***.println("Now do so a second time");

System.***out***.println("");

}

**private** **static** **void** contributionFromTyler() {

System.***out***.println("Begin by defining two arrays");

System.***out***.println("");

}

**private** **static** **void** ContributionFromConner() {

System.***out***.println("");

System.***out***.println("The merge method will now combine the arrays.");

System.***out***.println("");

}

//sorting method for input array

**public** **static** **int**[] sort(){

**int** n, temp;

Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Enter the size of your array:");

System.***out***.println("");

n = s.nextInt();

**int** a[] = **new** **int**[n];

System.***out***.println("Enter all the elements:");

System.***out***.println("");

**for** (**int** i = 0; i < n; i++)

{

a[i] = s.nextInt();

}

**for** (**int** i = 0; i < n; i++)

{

**for** (**int** j = i + 1; j < n; j++)

{

**if** (a[i] > a[j])

{

temp = a[i];

a[i] = a[j];

a[j] = temp;

}

}

}

System.***out***.print("Sorted from smallest to largest: ");

**for** (**int** i = 0; i < n - 1; i++)

{

System.***out***.print(a[i] + ", ");

}

System.***out***.print(a[n - 1]);

System.***out***.println("");

**return** a;

}

//merge method for both arrays

**public** **static** **int**[] merge(**int**[] a, **int**[] b) {

**int**[] answer = **new** **int**[a.length + b.length];

**int** i = a.length - 1, j = b.length - 1, k = answer.length;

**while** (k > 0)

answer[--k] =

(j < 0 || (i >= 0 && a[i] >= b[j])) ? a[i--] : b[j--];

**return** answer;

}

}

This code begins by importing the utilities and initializing two array variables outside of the main method (I originally thought that I couldn’t get the program to work without doing this, but later realized these variables weren’t essential at all. They’ve only been left in this report since this is how I uploaded it to GitHub and therefore this version is what my teammates had to work with, and I don’t want to change it up after the fact since the code still works regardless.) Once inside the main method, the two array variables are assigned the value of a called-sorting method. The sort()method itself is where the user inputs the desired size of the array, and then the integers within it. Once this data is obtained, the array is sorted in ascending order to be printed back to the user. After this I invoked the next() method in order to give a break in between the input sequences. The Sort() method is called once again, assigned to the second array variable. My fellow teammates contributed to this code by adding in their own simple methods containing print statements which aid in the flow of information in this area of the code. Once all of the data is gathered, sorted, and shown back to the user, a third array variable is then assigned to the merge() method, with the final values of the first two array variables as arguments within it. The two arrays are then merged together and sorted in full in ascending order, and the final result is returned to be printed back to the user.

The output of the code is as follows:

Begin by defining two arrays

Enter the size of your array:

3

Enter all the elements:

3 5 1

Sorted from smallest to largest: 1,3,5

Now do so a second time

Enter the size of your array:

5

Enter all the elements:

2 7 -4 9 0

Sorted from smallest to largest: -4,0,2,7,9

The merge method will now combine the arrays.

The merged lists are now:

[-4, 0, 1, 2, 3, 5, 7, 9]

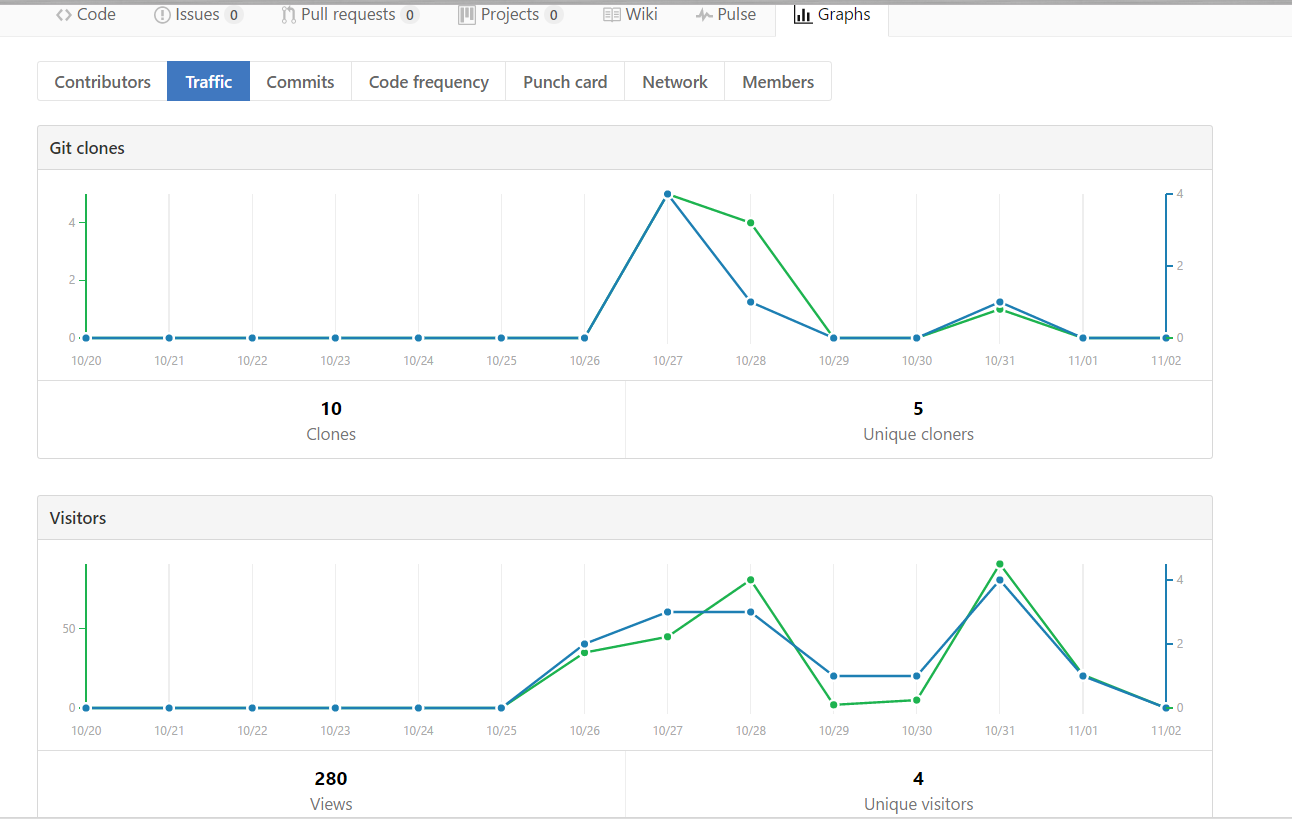
**Discussion**

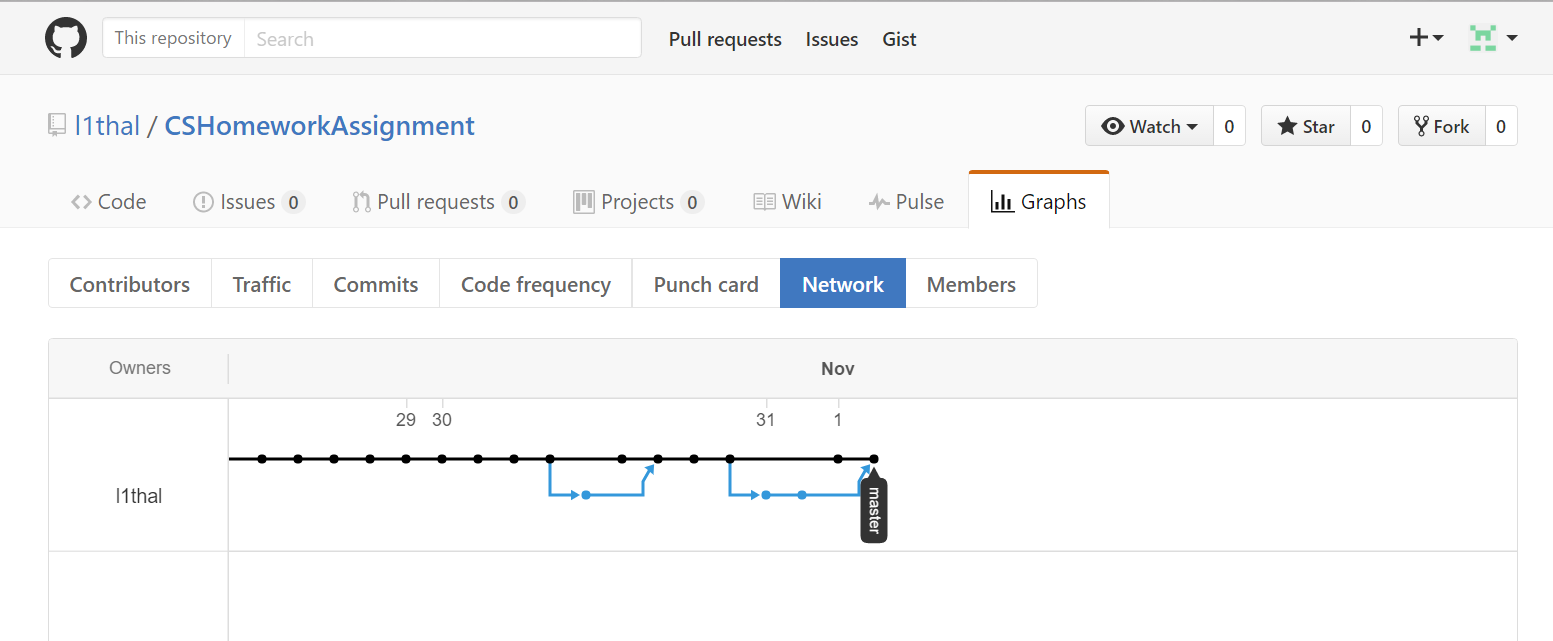
This project was likely the most informative one I have taken on so far. I learned how to use GitHub from a fellow team member, and will likely be utilizing this tool from now on. This was really my first run in using it practically to achieve a goal, and I now see why it is so valued. This code itself took a lot of headache for me to get right. Even with the help of the work of others as a backbone for what I was doing, it took me hours upon hours to figure out how to make it all come together. As per usual, it was lacking only a few minor tweaks in the manner that the merge() method was actually called. I hope to be able to do more projects like this and get more in depth with my contributions to teammates.

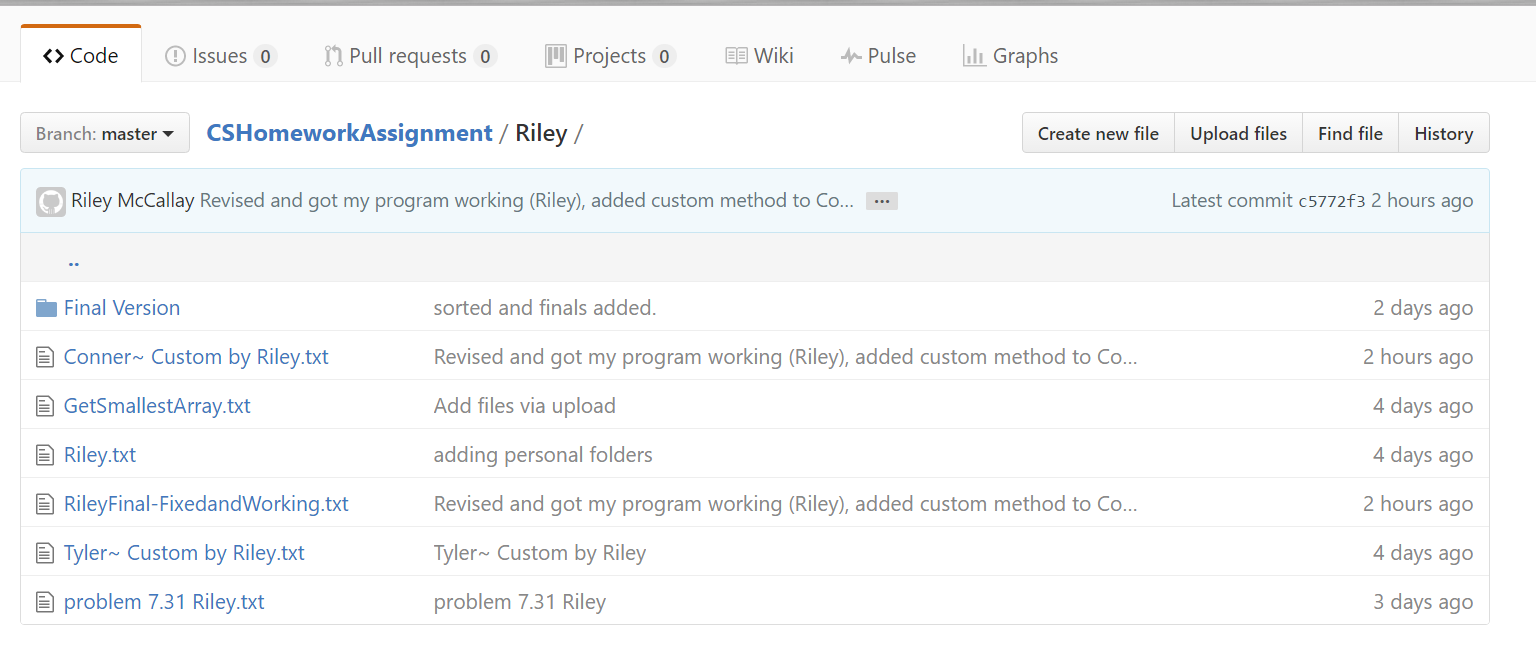
As far as teamwork itself, we all were able to do everything we needed and obtain each other’s code for our own personal tinkering quite easily. After we got everything set up initially, everything else was a breeze. For specifics, I contributed to Connor’s code by adding a simple method for one of his program’s main print statements. With Tyler’s, I put his code that worked out and displayed all of the possible combinations within a method I created called combo(). On a personal level, my group seemed to get along and work together well and everybody was very helpful and willing to participate. If I were to do anything differently next time, it would be to have a sit down discussion with the team prior to anything to plan things out a little more efficiently, regarding what project each person will do and how we will write methods for each other. That being said, it was hardly an issue, as we accomplished everything fairly well.

**GitHub Usage**

These are screenshots of the Graphs tab for proof of GitHub activity.







**References**

I used another person’s completed code I found online as the backbone of my Merge() method, the rest I did on my own, with the exception of my teammates’ contributions. The code for the merging was presented by user ShitalShah on *stackoverflow.com*, at the link below:

<http://stackoverflow.com/questions/5958169/how-to-merge-two-sorted-arrays-into-a-sorted-array>