

## **Programming Assignment 1 Report**

This programming assignment has the goal to recursively thread an array with the status of bulbs, to find the location of defective bulbs in the circuit and the number of threads that have been created for this purpose.

The code attached to this file divides recursively the series of bulbs into subseries and keeps investigating the subseries that have at least a defective bulb (i.e. any subseries that have at least one 0 in it), in order to find the defective bulb.

The solution to this problem gets an input text file, where the first line is the total number of light bulbs in the circuit and the following lines are the status of each bulb. We assume that a '0' indicates that the bulb is defective and that a '1' indicates that the bulb is functioning properly.

The first method of the two in the attached code is the main method (main()). The main method asks for the name of the input file with the information about the circuit and makes sure it is correct. If the filename is correct, the main method opens the file and reads it. It saves the first line into an Integer variable, this is the total number of bulbs in the circuit. The following lines are saved into an array of integers, this is the status of all the bulbs.

Then a thread is created that calls the method FindDefective(int[] bulbstatus, int numberOfBulbs, int start) with the bulbstatus array and numberOfBulbs variable filled by the text file, the start variable is the start index of the array, in this case, it is 0. The thread is started and then the main method waits until the thread is finished using .join(). Once the thread is finished, the method prints the number of threads used and the location of the defective bulbs in the circuit.

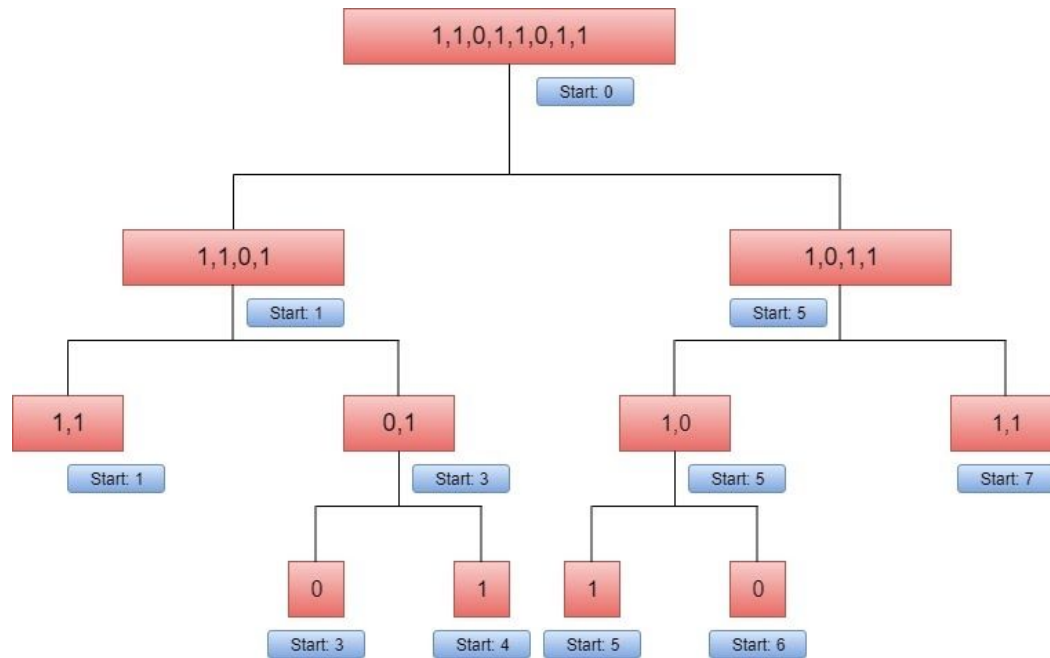
The second method of the code is the FindDefective method (FindDefective(int[] bulbstatus, int numberOfBulbs, int start)), the method gets an array of integers that hold the states of the light bulbs, the total number of light bulbs and the start index of the array given, this is used to keep track of the position of bulbs in the initial array.

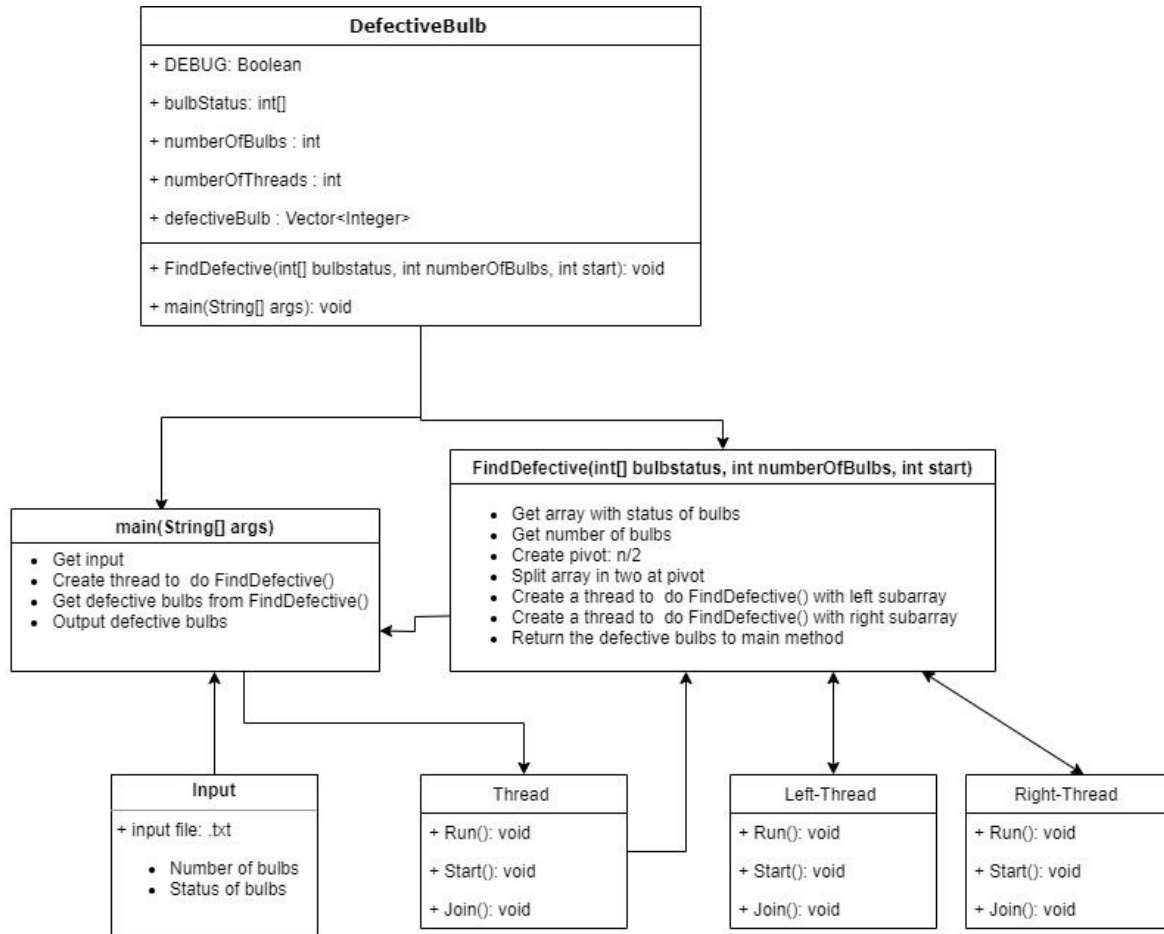
This method first checks if the array given to it has at least one defective bulb if there is no defective bulb, nothing is done and the threading is finished.

If a defective bulb is found, a pivot is created in the middle of the array by dividing the number of bulbs by 2 and saving it in a variable called "pivot", then two subarrays are created and initialized. One subarray for the left side of the array of size of "the pivot" and another for the right side of the array of size of "the number of bulbs minus the pivot", i.e. splitting the array into 2 subarrays at the pivot.

Once the subarrays are created they are filled from the initial array given to the method. Then two threads are created that call the method FindDefective, one with the left subarray and another with the right subarray, the numberOfBulbs variable is the size of each subarray and the start variable of the thread for the left subarray is the initial start given to the method and the start of the thread for the right subarray is the “pivot + the initial start” given to the method. The threads are started and then waits until both the threads are finished. The array is split into multiple arrays until there is no defective bulb in the subarray or until there's only the defective bulb in the subarray.

If the array has a defective bulb and it has a size of 1, then the recursive multi-threading is done for this branch and the location of a defective bulb is found and printed out. The location of this bulb is also saved to a vector variable to be printed by the main method when all threads are finished. Control is returned to the main method and the location of defective bulbs is printed.





### Conclusion:

In this programming assignment, we learned how to use threading to accomplish a task at a faster speed, since different parts of a program can run concurrently. We learned that multithreading is one way to solve an asynchrony problem. This was my first time dealing with threads while programming, therefore it took more time than expected to finish the code because I wanted to make sure I understand how threading works. This assignment made us learn how threading works and how it can be implemented in multiple tasks in an efficient way.

An example application of threading is text editors. When a text editor is used multiple things seem to happen at once. The user will be typing, while the text editor runs spell-checking, formatting of text and saving the text are done concurrently by multiple threads in the background.