**For Ques1**

In the first question, I have made two classes and have extended the thread class in those two classes; one handles the odd entry and executes till it does find that the array is sorted, and in the 2nd class, even is handled the same as odd ones then for sorting we need to start the two threads along and for taking the input I have taken an array list to do so one array list stores the size and other one stores the float elements that are needed to be sorted. And the time has been shown with the help of the java inbuilt program, which shows the time in nanoseconds. But the time here shown is coming greater when used threads and normally it is coming to be less.

**For Ques2**

In question 2, we have used the java collection framework that is treeset which is based on the red-black tree of which the AVL tree is a subset, so this root node remains the same. We have just created the threads to insert the elements faster parallel on the same set in two threads. We have used the two threads, and in the four threads part, we have again created four instances them starting them so that the elements can be added parallel among them and the time seen is coming less for the four threads many bits much for the two and more for the normal one times have been printed using the java inbuilt program.