Implementation of AVL-tree

You are the accountant of a big company and have been handed the task of maintaining a database of salaries of all the people working in the company. Whenever a person joins or leaves the company, you will need to update the database. Also, at times the manager may ask you for some statistics on the data. Multiple employees could have the same salary. You need to design and implement a data structure which supports the following operations (note x and y are positive integers) :

$\boldsymbol{A}(\boldsymbol{x})$: Add salary x to the database

$\boldsymbol{R}(\boldsymbol{x})$: Remove salary x from the database if it exists, otherwise return the salary closest to x

$\boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{y})$: Return the number of employees having salary in the range $[x, y]$

Max: Return the maximum salary

Min: Return the minimum salary

If the current number of distinct salaries in the company is n, queries A, R, Q should not take more than $O(log\ n)$ time. All the salaries are integers in the range $[1,10^6]$. You should initialize the database by reading salaries from a file. The first line of the file is an integer which specifies the number of data points. Subsequent lines contain the salaries (one on each line).

The test cases will consist of a sequence of the operations, A, R, Q, Max, Min and after each operation you should report the relevant information.