Short Take-Home Exam 3

(Deadline: 26th of May 2023, Friday, 23:59)

1 Problem

A group of n students (identified with ID numbers from 1 to n) take two exams (exams A and B) and get a *unique* score in each. The company that conducted the exams wants to prepare a certificate for each student, mentioning the rank of the student in the certificate. To boost student morale, the *rank* of a student is determined as

The number of students that achieved a higher score in both exams + 1

Notice that it is possible for two students to have the same rank.

Your job is to help the company to determine ranks of the students.

2 Input Format

```
N
A_1 B_1
...
A_N B_N
```

`N` = The number of students.
$\texttt{A}_i$ = The score of the student with the ID number i in exam A.
$\texttt{B}_i$ = The score of the student with the ID number i in exam B.

3 Output Format

```
R_1
R_2
...
R_N
```

$\texttt{R}_i$ = The rank (as defined above) of the student with ID number i.

4 Limits

- $1 \leq \texttt{N} \leq 100\,000$
- All exam scores ($\texttt{A}_i$ and $\texttt{B}_i$) are integers in the range [1, 2 000 000 000].
- Time Limit: 1 second
- Memory Limit: 128 MB
- Stack Limit: 8 MB

5 Clarifications

- Your solution is expected as a C++ program source named `sthe3.cpp` that reads from the standard input and writes to the standard output.
- It is OK to copy code from the sample codes we shared in ODTÜClass or from your individual previous THE submissions. Copying from elsewhere will be considered cheating.

- You are supposed to submit your code via the VPL item in ODTÜClass. Use the "evaluate" feature to run the auto-grader on your submission.
- The grade from the auto-grader is not final. We can do further evaluations and adjust your grade. Submissions that do not attempt a "reasonable solution" to the given task may lose points.
- Your code will be compiled with the options: `-std=c++2a -O3 -lm -Wall -Wextra -Wpedantic`
- Late submissions are not allowed.

6 Sample Input/Output Pairs

Input	Output
4	
6 7	1
2 1	4
3 4	3
5 8	1

Input	Output
8	
20 26	2
11 19	4
22 18	2
16 10	7
30 28	1
29 13	2
21 12	4
17 27	2

Input	Output
10	
55 90	2
17 41	8
60 70	4
12 15	10
85 95	1
83 25	3
43 75	4
73 51	3
89 72	1
68 76	2

7 Hints

- Observe that the scores are unique within each exam. You do NOT have to employ any kind of tie-breaking in this THE.
- We suggest that you treat one of the exam scores as "time" and then use the other score in a data structure to conduct queries/updates.
- We suggest using a kind of Fenwick tree to solve this problem. Of course, you are free to construct any data structures you want as long as you efficiently solve the problem.
- A well-implemented $O(n \log n)$-time solution is expected to get full marks.
- Notice that the exams scores can be pretty high. You will likely need a rank-space reduction (or something equivalent that avoids the negative effects of high scores).
- If you are unable to do rank-space reduction, you can still get significant partial points. 80% of the grading inputs will guarantee to have all exam scores to be $\leq 1\,000\,000$.
- You can use the standard library `std::sort` function to sort your data. You can sort the same data in different ways by providing a specific comparison function. For instance, to perform a descending sort on a `std::vector` with respect to a field of its elements:

```
std::sort(myVector.begin(),
          myVector.end(),
          [] (const auto & x, const auto & y) {
              return x.myField > y.myField;
          });
```

- As usual, please code the fastest algorithm you can find to get partial points.