Indian National Olympiad in Informatics, 2024

Question paper

Monsters

Monsters are magical creatures endowed with special powers, and they like to battle each other. Each Monster is of one of three **types**: Fire, Water or Grass. When two Monsters **battle** each other, one of them will win according to the following rules:

- A Fire Monster will always defeat a Grass Monster.
- A Water Monster will always defeat a Fire Monster.
- A Grass Monster will always defeat a Water Monster.
- If two Monsters of the same type battle each other, it is possible for **either one** of them to win.

There are `N` Monsters numbered from `1` to `N`. You are given the type of each Monster `i`. You wish to perform `Q` thought experiments, where the `j`-th thought experiment is the following:

- Imagine that the Monsters $L_j, L_j+1, L_j+2 \dots R_j$ (where $1 \le L_j \le R_j \le N$) are standing in a line from left to right, in order of number. Two Monsters are considered **adjacent** if there is no Monster standing between them. Any two **adjacent** Monsters may **battle** each other, and the Monster who **loses** the battle will leave the line. Notice that this may cause two Monsters in the line to start being adjacent. Eventually, after some number of battles, there will only be **one** Monster remaining. You wish to find the number of Monsters who could be the last remaining Monster. That is, you wish to count the number of Monsters `i` such that there is **at least one** valid sequence of battles and outcomes of battles by which `i` is the last remaining Monster.

Please note that no thought experiment is impacted in any way by any other thought experiment.

Input Format

The first line contains a single integer `N`, the number of Monsters.

The second line contains a string `A` of `N` characters without spaces, where the `i`-th character is `F`, `W`, or `G`, depending on whether the `i`-th Monster is Fire type, Water type, or Grass type, respectively.

The third line contains a single integer `Q`, the number of thought experiments to be performed.

The next `Q` lines describe these thought experiments. The `j`-th of these lines contains two space-separated integers L_j and R_j.

Output Format

You should print `Q` lines of output. The `j`-th of these lines should consist of a single integer, which is the number of Monsters that could be the last remaining Monster in thought experiment `j`.

Scoring

The test data for this problem is divided into multiple subtasks. In order to pass a subtask, your submitted program must solve every test case within that subtask correctly and within the time and memory limits.

You will be awarded the points allocated to a subtask if **at least one** submission you make during the contest passes that subtask. You do **not** need to combine your solutions for multiple subtasks into a single submission.

Please keep in mind that the subtasks are not necessarily arranged in increasing order of difficulty. We encourage you to try as many subtasks as possible.

Constraints

In all test data, it is guaranteed that:

- $1 \le N \le 10^5$.
- A_i = F, W or G, for all $1 \le i \le N$.
- $1 \le Q \le 10^5$.
- $1 \le L_j \le R_j \le N$, for all $1 \le j \le Q$.

Subtasks

Constraint for Subtasks 1-9
You only have to perform one thought experiment, and this thought experiment covers all the Monsters. In other words, $Q = 1$, $L_1 = 1$, $R_1 = N$.

- **Subtask 1** (4 points) There are only Fire Monsters.
- **Subtask 2** (5 points) There are no Grass Monsters.
- **Subtask 3** (7 points) $N = 3$.
- **Subtask 4** (5 points) $N \le 35$. Further, there are at most 2 pairs of adjacent Monsters who are of different types.
- **Subtask 5** (11 points) $N \le 35$.
- **Subtask 6** (10 points) $N \le 80$.
- **Subtask 7** (10 points) $N \le 400$.
- **Subtask 8** (10 points) $N \le 1500$.
- **Subtask 9** (16 points) $N \le 10^5$.

Constraint for Subtasks 10-11
There may be multiple thought experiments.

- **Subtask 10** (6 points) There are no Grass Monsters.
- **Subtask 11** (16 points) No additional constraints.

Sample 0

Input

```
7
FFWGWFG
2
1 2
2 4
```

Output

```
2
2
```

Explanation

In this sample, there are two thought experiments.

Consider the first thought experiment. Initially, the line contains Monsters `[1, 2]`. Both of them are Fire type, so in a battle between them, it is possible for either one to win. Therefore, both Monsters `1` and `2` can be the last remaining Monster and the answer is `2`.

Consider the second thought experiment. Here, the line initially contains Monsters `[2, 3, 4]`, and they have types Fire, Water and Grass respectively. It is possible for Monsters `2` and `4` to be the last remaining Monster, so the answer is `2`.

Here is one possible sequence of battles by which Monster `2` can be the last remaining Monster:

- Initially, the line contains all the Monsters `[2, 3, 4]`. Monster `3` and Monster `4` can battle. As Monster `3` is of Water type and Monster `4` is of Grass type, Monster `4` will win and Monster `3` will leave the line.
- Now, the remaining Monsters in the line are `[2, 4]`. Monster `2` and Monster `4` can battle. As Monster `2` is of Fire type and Monster `4` is of Grass type, Monster `2` will win and Monster `4` will leave the line.
- Monster `2` is the last remaining Monster.

Here is one possible sequence of battles by which Monster `4` can be the last remaining Monster:

- Initially, the line contains all the Monsters `[2, 3, 4]`. Monster `2` and Monster `3` can battle. As Monster `2` is of Fire type and Monster `3` is of Water type, Monster `3` will win and Monster `2` will leave the line.
- Now, the remaining Monsters in the line are `[3, 4]`. Monster `3` and Monster `4` can battle. As Monster `3` is of Water type and Monster `4` is of Grass type, Monster `4` will win and Monster `3` will leave the line.
- Monster `4` is the last remaining Monster.

There is no such sequence of battles by which Monster `3` can be the last remaining Monster.

Subtask Validity

Sample 0 is valid for subtask 11.

Sample 1

Input

```
5
FWGFW
1
1 5
```

Output

```
4
```

Explanation

In this example, there is only one thought experiment which covers all the monsters. It is possible for the Monsters `1`, `2`, `3` and `5` to be the last remaining Monster, so the answer is `4`.

Here is one possible sequence of battles by which Monster `1` is the last remaining Monster:

- Initially, the line contains all the Monsters `[1, 2, 3, 4, 5]`. Monster `3` and Monster `2` can battle. As Monster `3` is of Grass type and Monster `2` is of Water type, Monster `3` will win and Monster `2` will leave the line.
- Now, the remaining Monsters in the line are `[1, 3, 4, 5]`. Monster `4` and Monster `5` can battle. As Monster `4` is of Fire type and Monster `5` is of Water type, Monster `5` will win and Monster `4` will leave

the line.

- Now, the remaining Monsters in the line are `[1, 3, 5]`. Monster `3` and Monster `5` can battle. As Monster `3` is of Grass type and Monster `5` is of Water type, Monster `3` will win and Monster `5` will leave the line.
- Now, the remaining Monsters in the line are `[1, 3]`. Monster `1` and Monster `3` can battle. As Monster `1` is of Fire type and Monster `3` is of Grass type, Monster `1` will win and Monster `3` will leave the line.
- Monster `1` is the last remaining Monster.

Similar to this, for each Monster `2`, `3` or `5` it is possible to find such a sequence of battles and outcomes of battles such that that Monster will be the last remaining Monster. For Monster `4`, it is impossible to find such a sequence.

Subtask Validity

Sample 1 is valid for subtasks 5, 6, 7, 8, 9 and 11.

Sample 2

Input

```
35
FWFWFWWWFWWFWWFFWFWWFWWWWFWFWWWFWWW
1
1 35
```

Output

```
23
```

Subtask Validity

Sample 2 is valid for subtasks 2, 5, 6, 7, 8, 9, 10 and 11.

Fertilizer

You are a farmer. You own `F` **fields** of land, which are arranged in a straight line. They are numbered `1` to `F` from left to right.

You have recently acquired a crop duster, which is an airplane that can quickly spray fertilizer over a large amount of land. The crop duster is preprogrammed with `N` possible **trips**, numbered from `1` to `N`. In trip `i`, the crop duster flies over the fields $L_i, L_i+1, L_i+2 \ldots R_i$, where $1 \le L_i \le R_i \le F$, and sprays fertilizer over all these fields.

You now plan to use a consecutive set of preprogrammed trips. You would like to answer `Q` questions, where the `j`-th question is the following:

- If the crop duster makes only the trips numbered $X_j, X_j+1, X_j+2 \ldots Y_j$, where $1 \le X_j \le Y_j \le N$, how many fields will be fertilized **at least once**?

Note that each question is independent from the other questions, i.e. we do not actually carry out the trips.

Input Format

The first line contains a single integer `F`, the number of fields.

The second line contains a single integer `N`, the number of preprogrammed trips.

The next `N` lines describe the trips. The `i`-th of these lines contains two integers L_i and R_i.

The next line contains a single integer `Q`, the number of questions you wish to answer.

The next `Q` lines describe the questions. The `j`-th of these lines contains two integers X_j and Y_j.

Output Format

You should print `Q` lines of output. The `j`-th line should consist of a single integer, the answer to the `j`-th question.

Scoring

The test data for this problem is divided into multiple subtasks. In order to pass a subtask, your submitted program must solve every test case within that subtask correctly and within the time and memory limits.

You will be awarded the points allocated to a subtask if **at least one** submission you make during the contest passes that subtask. You do **not** need to combine your solutions for multiple subtasks into a single submission.

Please keep in mind that the subtasks are not necessarily arranged in increasing order of difficulty. We encourage you to try as many subtasks as possible.

Constraints

In all test data, it is guaranteed that:

- $1 \le F \le 10^6$.
- $1 \le N \le 5 * 10^5$.
- $1 \le L_i \le R_i \le F$, for all $1 \le i \le N$.

- $1 \le Q \le 10^6$.
- $1 \le X_j \le Y_j \le N$, for all $1 \le j \le Q$.

Subtasks

- **Subtask 1** (4 points) $F, Q, N \le 100$.
- **Subtask 2** (6 points) $F, Q, N \le 2000$. Further, $X_j = 1$ for all $1 \le j \le Q$.
- **Subtask 3** (8 points) $F, Q, N \le 2000$.
- **Subtask 4** (5 points) $N \le 10^5$. Further, $R_i < L_{i+1}$ for all $1 \le i < N$.
- **Subtask 5** (8 points) $N \le 10^5$. Further, $L_i < L_{i+1}$ and $R_i < R_{i+1}$ for all $1 \le i < N$.
- **Subtask 6** (10 points) $N, Q \le 2 * 10^4$.
- **Subtask 7** (8 points) $N, Q \le 5 * 10^4$. Further, $L_i = R_i$ for all $1 \le i \le N$.
- **Subtask 8** (15 points) $N \le 10^5$. Further, $X_j \le X_{j+1}$ and $Y_j \le Y_{j+1}$ for all $1 \le j < Q$.
- **Subtask 9** (17 points) $N \le 10^5$.
- **Subtask 10** (19 points) No additional constraints

Sample 0

Input

```
5
3
1 4
2 3
4 5
4
1 2
2 3
1 3
1 1
```

Output

```
4
4
5
4
```

Explanation

In Sample 0, there are 3 trips : Trip 1 covering fields 1 to 4, trip 2 covering fields 2 to 3 and trip 3 covering fields 4 to 5.

- In question 1, the trips 1, and 2 are under consideration. Trip 1 will cover the fields 1, 2, 3, 4. Trip 2 will cover the fields 2, 3. The fields 1, 2, 3, 4 are fertilized at least once, so the answer is 4.
- In question 2, the trips 2, and 3 are under consideration. Trip 2 will cover the fields 2, 3. Trip 3 will

cover the fields `4`, `5`. The fields `2`, `3`, `4`, `5` are fertilized at least once, so the answer is `4`.

- In question `3`, all the trips `1`, `2`, and `3` are under consideration. Trip `1` will cover the fields `1`, `2`, `3`, `4`. Trip `2` will cover the fields `2`, `3`. Trip `3` will cover the fields `4`, `5`. All the fields `1`, `2`, `3`, `4`, `5` are fertilized at least once, so the answer is `5`.
- In question `4`, only trip `1` is under consideration. Trip `1` will cover the fields `1`, `2`, `3`, `4`. All the fields `1`, `2`, `3`, `4` are fertilized at least once, so the answer is `4`.

Subtask Validity

Sample 0 is valid for subtasks 1, 3, 6, 9 and 10.

Sample 1

Input

```
10
4
1 4
2 5
7 8
8 10
4
1 4
1 2
4 4
3 4
```

Output

```
9
5
3
4
```

Subtask Validity

Sample 1 is valid for subtasks 1, 3, 5, 6, 9 and 10.

Sample 2

Input

```
10
5
4 8
2 2
1 3
5 7
9 10
5
1 1
1 2
1 3
1 4
1 5
```

Output

```
5
6
8
8
10
```

Subtask Validity

Sample 2 is valid for subtasks 1, 2, 3, 6, 9 and 10.

Sample 3

Input

```
10
5
1 4
2 10
3 6
5 8
2 6
5
1 2
3 4
3 5
4 5
5 5
```

Output

```
10
6
7
7
5
```

Subtask Validity

Subtask 3 is valid for subtasks 1, 3, 6, 8, 9 and 10.

Sample 4

Input

```
10
7
1 4
2 3
4 7
5 10
3 5
6 8
1 2
5
1 7
2 5
3 4
6 7
1 3
```

Output

```
10
9
7
5
7
```

Subtask Validity

Sample 4 is valid for subtasks 1, 3, 6, 9 and 10.

Trees

A **tree** is a set of `N` **nodes** and `N-1` undirected **edges**. Each edge connects two distinct nodes, and between any two nodes there is a unique path that doesn't repeat edges. Further, each edge has an integer **edge weight**.

You are given two trees, **tree X** and **tree Y**. In both these trees, there are `N` nodes numbered from `1` to `N` and `N-1` edges numbered from `1` to `N-1`, but the specific edge structures and edge weights of tree X and tree Y may differ.

In tree X, the `i`-th edge connects nodes A_i and B_i and has weight C_i.

In tree Y, the `j`-th edge connects nodes U_j and V_j and has weight W_j.

For some two nodes `p` and `q` such that $1 \le p < q \le N$, $cost_X(p, q)$ is defined as the **largest** edge weight that occurs on the unique path between `p` and `q` in tree X. Similarly, $cost_Y(p, q)$ is defined as the **largest** edge weight that occurs on the unique path between `p` and `q` in tree Y.

You are asked to find the number of pairs of nodes `(p, q)` such that $1 \le p < q \le N$ and $cost_X(p, q) \le cost_Y(p, q)$. (In total, there are `N(N-1)/2` pairs of nodes `(p, q)` such that $1 \le p < q \le N$.)

Input Format

The first line of input contains a single integer `N`, the number of nodes.

The next `N-1` lines describe tree X. The `i`-th of these lines contains three integers, A_i, B_i, and C_i.

The next `N-1` lines describe tree Y. The `j`-th of these lines contains three integers, U_j, V_j, and W_j.

Output Format

You should print a single integer, the number of pairs of nodes `(p, q)` such that $1 \le p < q \le N$ and $cost_X(p, q) \le cost_Y(p, q)$.

Scoring

The test data for this problem is divided into multiple subtasks. In order to pass a subtask, your submitted program must solve every test case within that subtask correctly and within the time and memory limits.

You will be awarded the points allocated to a subtask if **at least one** submission you make during the contest passes that subtask. You do **not** need to combine your solutions for multiple subtasks into a single submission.

Please keep in mind that the subtasks are not necessarily arranged in increasing order of difficulty. We encourage you to try as many subtasks as possible.

Constraints

In all test data, it is guaranteed that:

- $1 \le N \le 10^5$.
- For all `i` such that $1 \le i \le N-1$:

- $1 \le A_i \le N$.
- $1 \le B_i \le N$.
- $A_i \ne B_i$.
- $1 \le C_i \le 10^9$.

- For all j such that $1 \le j \le N-1$:
  - $1 \le U_j \le N$.
  - $1 \le V_j \le N$.
  - $U_j \ne V_j$.
  - $1 \le W_j \le 10^9$.

Please be aware that the output for this problem may not fit in 32-bit integers. You may need to use 64-bit integers in your computations.

Notation

The following notation is used to describe the subtasks:

- The *Lines* constraint is present in subtasks 1, 4 and 5. This means that for all i such that $1 \le i \le N-1$, the following conditions hold:
  - $A_i = i$.
  - $B_i = i+1$.
  - $U_i = i$.
  - $V_i = i+1$.
- The constraint $C_1 \le C_2 \le ... \le C_{N-1}$ is present in subtasks 1, 4, 6 and 7. This means that for each i such that $1 \le i \le N-2$, $C_i \le C_{i+1}$ holds.
- The constraint $W_1 \le W_2 \le ... \le W_{N-1}$ is present in subtasks 1, 4, 6 and 7. This means that for each j such that $1 \le j \le N-2$, $W_j \le W_{j+1}$ holds.
- The constraint $(A_i, B_i) = (U_i, V_i)$ is present in subtasks 6 and 7. This means that for all i such that $1 \le i \le N-1$, $A_i = U_i$ and $B_i = V_i$. In other words, tree X and tree Y have the same $N-1$ edges (but may have differing edge weights).

Subtasks

- **Subtask 1** (4 points) $N \le 200$, *Lines*, $C_1 \le C_2 \le ... \le C_{N-1}$, $W_1 \le W_2 \le ... \le W_{N-1}$.
- **Subtask 2** (5 points) $N \le 200$.
- **Subtask 3** (8 points) $N \le 2000$.
- **Subtask 4** (9 points) *Lines*, $C_1 \le C_2 \le ... \le C_{N-1}$, $W_1 \le W_2 \le ... \le W_{N-1}$.
- **Subtask 5** (15 points) *Lines*.
- **Subtask 6** (10 points) $(A_i, B_i) = (U_i, V_i)$, $C_1 \le C_2 \le ... \le C_{N-1}$, all W_j are equal.
- **Subtask 7** (14 points) $(A_i, B_i) = (U_i, V_i)$, $C_1 \le C_2 \le ... \le C_{N-1}$, $W_1 \le W_2 \le ... \le W_{N-1}$.
- **Subtask 8** (35 points) No additional constraints.

Sample 0

Input

```
4
1 2 2
2 3 3
3 4 9
1 2 2
2 3 6
3 4 8
```

Output

```
3
```

Explanation

In this sample, there are `N = 4` nodes, and there are `6` pairs of nodes in total, namely `(1, 2)`, `(1, 3)`, `(1, 4)`, `(2, 3)`, `(2, 4)` and `(3, 4)`.

Out of these, the pairs `(p, q) = (1, 2)`, `(2, 3)`, `(1, 3)` all satisfy the condition $\text{cost}_X(p, q) \le \text{cost}_Y(p, q)$, and no other pairs satisfy the condition. So, the answer for this sample is `3`.

- Consider the pair `(p, q) = (1, 3)`.
  In tree X, there are two edges between `1` and `3`, them being the edge connecting nodes `1` and `2` with weight `2` and the edge connecting nodes `2` and `3` with weight `3`. Thus, the maximum weight is `3` and $\text{cost}_X(1, 3) = 3$.
  In tree Y, there are two edges between `1` and `3`, them being the edge connecting nodes `1` and `2` with weight `2` and the edge connecting nodes `2` and `3` with weight `6`. Thus, the maximum weight is `6` and and $\text{cost}_Y(1, 3) = 6$.
  This means that $\text{cost}_X(p, q) \le \text{cost}_Y(p, q)$ is satisfied and the pair `(1, 3)` should be counted in the answer.
- Consider the pair `(p, q) = (2, 4)`. $\text{cost}_X(2, 4) = 9$ and $\text{cost}_Y(2, 4) = 8$. As the condition $\text{cost}_X(p, q) \le \text{cost}_Y(p, q)$ is not satisfied, the pair `(2, 4)` should not be counted in the answer.

Subtask Validity

Sample 0 is valid for subtasks 1, 2, 3, 4, 5, 7, 8.

Sample 1

Input

```
5
1 2 8
2 3 6
3 4 2
4 5 7
1 2 10
2 3 4
3 4 7
4 5 5
```

Output

```
8
```

Subtask Validity

Sample 1 is valid for subtasks 2, 3, 5 and 8.

Sample 2

Input

```
5
1 4 6
2 3 8
1 2 9
2 5 10
1 4 8
2 3 8
1 2 8
2 5 8
```

Output

```
2
```

Subtask Validity

Sample 2 is valid for subtasks 2, 3, 6, 7 and 8.

Sample 3

Input

```
6
1 2 2
2 4 6
3 2 8
4 5 8
1 6 10
1 2 3
2 4 5
3 2 5
4 5 9
1 6 11
```

Output

```
10
```

Subtask Validity

Sample 3 is valid for subtasks 2, 3, 7 and 8.

Sample 4

Input

```
8
1 2 6
2 3 6
1 4 10
2 5 7
4 6 7
3 7 1
5 8 4
1 2 2
1 3 6
1 4 2
4 5 9
1 6 2
5 7 1
7 8 1
```

Output

```
11
```

Subtask Validity

Sample 4 is valid for subtasks 2, 3 and 8.