Data Structures 1 (Beta)

April Camp 2022

Table of contents

1 Should you be here?

2 Problem 1: Dynamic Graph Connectivity (Offline)

3 Problem 2: New Home

Should you be here?

- Only come to the Beta lecture if you know the Alpha content.
- That is, you should understand basic range trees (aka segment trees). In particular:
  - You have solved and implemented Min Tree II
  - You have solved and implemented Store supplies
- In this lecture we'll look at some problems with interesting data structure solutions. The slides have the problems, and we will discuss the solutions together.

Dynamic Graph Connectivity (Offline): Statement

There is a graph with N vertices and no edges (initially). You should support Q of the following updates/queries:

- Add an undirected edge between vertices a and b (you are guaranteed this edge doesn't exist)
- Remove the edge between a and b (you are guaranteed this edge does exist)
- Query how many connected components are in the graph

The problem is **offline** meaning all updates and queries are given to you beforehand.

Constraints

- $N \leq 100\,000$
- $Q \leq 200\,000$

Dynamic Graph Connectivity (Offline): Sample

Sample Input

```
3 7
?
+ 1 2
?
+ 1 3
?
- 1 2
?
```

Sample Output

```
3
2
1
2
```

Explanation

There are $N = 3$ vertices and $Q = 7$ updates/queries.

1. In the first query there are no edges and so 3 connected components.
2. An edge is added between vertices 1 and 2, so there are two connected components.
3. An edge is added between vertices 1 and 3 so there is one connected component.
4. The edge between vertices 1 and 2 is removed, so there are two connected components.

New Home: Statement

There is a street with N stores on it, where the position of store i is x_i. Each store also has a type t_i (from 1 to K) and will be open from year a_i to year b_i (inclusive).

You must answer Q (offline) queries. In each query, you are given a location l_i and year y_i. Define the accessibility of store type t as the distance (from l_i) to the nearest store of type t that is open in the year y_i. The inconvenience is defined as the maximum accessibility of any store type. If all the store types are not available, then the inconvenience is -1. For each query, you must output the inconvenience.

Constraints

- $N, Q \leq 300\,000$.
- $1 \leq K \leq N$.
- $1 \leq x_i, a_i, b_i \leq 10^8$ for all i.
- $a_i \leq b_i$ for all i.

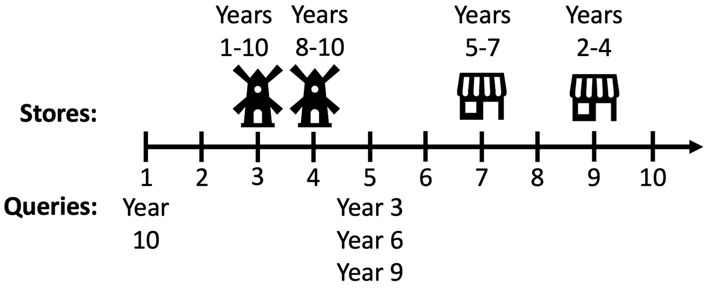
New Home: Sample

Input Format

The first line contains N, K and Q.

The next N lines describe the stores, with x_i, t_i, a_i and b_i.

The next Q lines describe the queries, with l_i and y_i.



Sample Input

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

Sample Output

```
4
2
-1
-1
```

New Home: Subtask 1

- $N, Q \leq 400$.
- Can solve in $O(NQ)$.

New Home: Subtask 2

- $N, Q \leq 60\,000$.
- $K \leq 400$
- Can solve in $O(QK \log(N))$.

New Home: Subtask 3

- $N, Q \leq 300\,000$.
- $a_i = 1, b_i = 10^8$ for all stores i. That is, all stores exist for all years.

New Home: Subtask 5

- $N, Q \leq 60\,000$.
- Can solve in $O(N\sqrt{N})$ or $O(N \log^2(N))$.

New Home: Subtask 6

- Full problem: $N, Q \leq 300\,000$.
- Can solve in $O(N \log(N))$.