1801. Number of Orders in the Backlog

My Submissions (/contest/weekly-contest-233/problems/number-of-orders-in-the-backlog/submissions/) Back to Contest (/contest/weekly-contest-233/)

User Accepted:	2251
User Tried:	3062
Total Accepted:	2294
Total Submissions:	5795
Difficulty:	Medium

You are given a 2D integer array `orders`, where each `orders[i] = [price_i, amount_i, orderType_i]` denotes that `amount_i` orders have been placed of type `orderType_i` at the price `price_i`. The `orderType_i` is:

- `0` if it is a batch of `buy` orders, or
- `1` if it is a batch of `sell` orders.

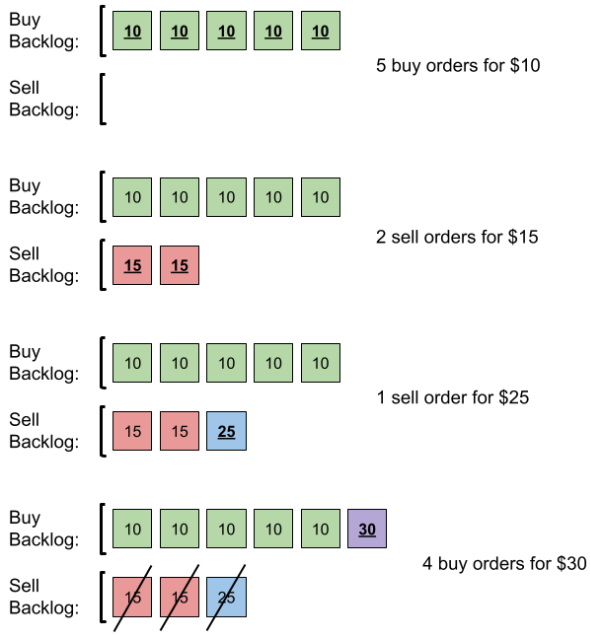
Note that `orders[i]` represents a batch of `amount_i` independent orders with the same price and order type. All orders represented by `orders[i]` will be placed before all orders represented by `orders[i+1]` for all valid `i`.

There is a **backlog** that consists of orders that have not been executed. The backlog is initially empty. When an order is placed, the following happens:

- If the order is a `buy` order, you look at the `sell` order with the **smallest** price in the backlog. If that `sell` order's price is **smaller than or equal to** the current `buy` order's price, they will match and be executed, and that `sell` order will be removed from the backlog. Else, the `buy` order is added to the backlog.
- Vice versa, if the order is a `sell` order, you look at the `buy` order with the **largest** price in the backlog. If that `buy` order's price is **larger than or equal to** the current `sell` order's price, they will match and be executed, and that `buy` order will be removed from the backlog. Else, the `sell` order is added to the backlog.

Return *the total **amount** of orders in the backlog after placing all the orders from the input*. Since this number can be large, return it **modulo** $10^9 + 7$.

Example 1:




```
Input: orders = [[10,5,0],[15,2,1],[25,1,1],[30,4,0]]
Output: 6
Explanation: Here is what happens with the orders:
- 5 orders of type buy with price 10 are placed. There are no sell orders, so the 5 orders are added to the backlog.
- 2 orders of type sell with price 15 are placed. There are no buy orders with prices larger than or equal to 15, so the 2 orde
- 1 order of type sell with price 25 is placed. There are no buy orders with prices larger than or equal to 25 in the backlog,
- 4 orders of type buy with price 30 are placed. The first 2 orders are matched with the 2 sell orders of the least price, whic
Finally, the backlog has 5 buy orders with price 10, and 1 buy order with price 30. So the total number of orders in the backlo
```

Example 2: