```java
public static int maximumSum(int[] A, int[][] ops) {
    int n = A.length;
    int[] pref = new int[n];

    for(int[] op:ops){
        int sp = op[0];
        int ep = op[1];
        pref[sp]++;
        if(ep+1<=n-1) pref[ep+1]--;
    }
    for(int i = 1;i<n;i++) pref[i] += pref[i-1];

    Arrays.sort(A);
    Arrays.sort(pref);
    int md = 1000000007;
    int ans = 0;
    for(int i = 0;i<n;i++){
        ans =( ans + (A[i] * pref[i])%md)%md;
    }
    return ans%md;
}
```

$$A = \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \\ 2 & 1 & 4 & 3 & 5 & 1 \end{array}$$

$$Pref = \quad 1 \quad 2 \quad 2 \quad 3 \quad 2 \quad 1$$

sp   ep
[1, 3]   A→  1 1 2 3 4 5
[2, 4]   P→  1 1 2 2 2 3
[4, 5]
[3, 3]   $1+1+4+6+8+15$
[0, 1)

---

⑥   ④   $10^5$   $\begin{bmatrix} 2 , & 1 , & 5 \\ 3 , & 4 , & 6 \end{bmatrix}$ $\begin{array}{l} sp++ = P \\ ep-- = P \end{array}$

$10^3$

$\begin{array}{cccccccc} 0 & 2 & 0 & 3 & 3 & -2 & 0 & -3 \\ 0 & 2 & 2 & 2 & 5 & 3 & 0 & 0 \end{array}$

$\begin{bmatrix} \end{bmatrix}$

2 1 5
3 3 7

$\begin{array}{cccccccc} 0 & 2 & 2 & 3 & 0 & -2 & 0 & -3 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$

---

5 ⑩   3 ①   4 ③   2   3 ⑤   1

5   6   9   13   15   18   17

① Create pref summary
② answer queries

$O(N) + O(Q)$
$\Rightarrow O$

$sp, ep \Rightarrow Pref[ep] - pref[sp-1]$
$O(1)$