

Bài 6. Tổng lớn nhất

💡 Bài toán

Cho một mảng số nguyên $A[]$ gồm N phần tử và Q truy vấn. Mỗi truy vấn yêu cầu tính tổng các phần tử từ chỉ số L đến R . Tuy nhiên, trước khi thực hiện các truy vấn, bạn được **phép hoán đổi lại thứ tự các phần tử trong mảng đúng 1 lần**.

Yêu cầu là sắp xếp lại mảng sao cho **tổng giá trị các truy vấn đạt lớn nhất**.

📁 Input

- Dòng 1: 2 số nguyên N và Q .
- Dòng 2: N số nguyên của mảng $A[]$.
- Q dòng tiếp theo: mỗi dòng gồm 2 số L và R .

📁 Output

- In ra một số nguyên duy nhất là **tổng lớn nhất có thể đạt được từ các truy vấn**, sau khi đã sắp xếp lại mảng.

🔍 Phân tích & Ý tưởng

- Mỗi truy vấn (L, R) yêu cầu tổng các phần tử từ L đến R . Nếu bạn sắp xếp mảng một cách khôn ngoan sao cho **những phần tử có tần suất xuất hiện trong truy vấn cao được đặt ở những chỉ số xuất hiện nhiều nhất**, thì tổng sẽ lớn nhất.
- ✅ Giải pháp:
 - Tính **tần suất xuất hiện** của từng vị trí trong mảng $A[]$ thông qua mảng cộng dồn hiệu.
 - Sắp xếp **tần suất các vị trí giảm dần**.
 - Sắp xếp mảng $A[]$ **tăng dần**.
 - Nhân từng phần tử lớn trong $A[]$ với vị trí có tần suất cao \rightarrow tối ưu tổng.

🧠 Ý tưởng cài đặt

- Dùng mảng $freq[]$ có kích thước $N + 2$, để đánh dấu hiệu ứng truy vấn bằng mảng hiệu.
- Duyệt qua Q truy vấn, với mỗi (L, R) :

```
freq[L - 1]++;  
freq[R]--;
```

- Duyệt qua mảng $freq$ để tính tổng tần suất tại mỗi vị trí.
- Sắp xếp $A[]$ tăng dần và $freq[]$ giảm dần \rightarrow Nhân từng cặp tương ứng để lấy tổng.

📁 Sample Input

```
6 2  
5 2 5 3 5 1
```

2 5
2 4

✓ Sample Output

33

Giải thích

- Tần suất vị trí: `[0, 0, 2, 2, 2, 1]`
- `A[]` sắp tăng: `[1, 2, 3, 5, 5, 5]`
- Tần suất giảm: `[2, 2, 2, 1, 0, 0]`
- Tổng lớn nhất: `5×2 + 5×2 + 5×2 + 3×1 + 2×0 + 1×0 = 33`

Độ phức tạp

- `O(N log N + Q)` do:
 - Tính tần suất: `O(Q)`
 - Tính mảng cộng dồn: `O(N)`
 - Sắp xếp: `O(N log N)`
 - Nhân và cộng kết quả: `O(N)`