

Promotion

Big-O Blue - Lecture 7: Heap

Tóm tắt đề bài

Tóm tắt đề bài

Chuỗi siêu thị Bytelandian tổ chức khuyến mãi cho khách hàng. Để tham gia chương trình, khách hàng sau khi mua sắm, có thể bỏ hóa đơn vào thùng phiếu.

- **Mỗi ngày**, siêu thị sẽ chọn ra 2 hóa đơn trong thùng phiếu: 1 hóa đơn có giá trị lớn nhất (tạm gọi là MAX) và 1 hóa đơn có giá trị nhỏ nhất (MIN). Khách hàng đã mua hóa đơn MAX sẽ được nhận một khoản tiền khuyến mãi tương đương với: $MAX - MIN$.
- Hóa đơn MAX và MIN sau khi được chọn sẽ **KHÔNG** được bỏ lại vào thùng.
- Đảm bảo luôn có ít nhất 2 hóa đơn trong thùng phiếu mỗi ngày.

Câu hỏi: Tính tổng số tiền siêu thị phải chi trả cho chương trình khuyến mãi này.

Input

Dòng đầu tiên là n ($1 \leq n \leq 5000$): số ngày diễn ra chương trình.

n dòng tiếp theo, mỗi dòng:

- Số đầu tiên là k ($1 \leq k \leq 10^5$), là số hóa đơn đã được bỏ vào thùng phiếu của ngày hôm đó.
- k số tiếp theo là các số nguyên không âm, tương ứng với giá trị của các hóa đơn (nhỏ hơn 10^6).

Tổng số hóa đơn được bỏ vào thùng phiếu trong suốt chương trình bé hơn 10^6 .

Output

Một dòng duy nhất, là tổng số tiền giải thưởng của toàn bộ chương trình.

Giải thích ví dụ

Ví dụ

Input:

5
3 1 2 3
2 1 1
4 10 5 5 1
0
1 2

Output:

19

Chương trình diễn ra trong 5 ngày.

Ngày đầu tiên có 3 hóa đơn được bỏ vào thùng: 1 2 3

Thùng hiện tại gồm: 1 2 3

→ MAX: 3, MIN: 1 → Số tiền giải thưởng: $3 - 1 = 2$

Trong thùng còn: 2

Ngày thứ hai có 2 hóa đơn: 1 1

Thùng hiện tại gồm: 1 1 2

→ MAX: 2, MIN: 1 → Số tiền: $2 - 1 = 1$

Trong thùng còn: 1

Ngày thứ ba có 4 hóa đơn: 10 5 5 1

Thùng hiện tại gồm: 1 1 5 5 10

→ MAX: 10, MIN: 1 → Số tiền: $10 - 1 = 9$

Trong thùng còn : 1 5 5

Ví dụ

Input:

5

3 1 2 3

2 1 1

4 10 5 5 1

0

1 2

Output:

19

Ngày thứ 4 không có hóa đơn mới

Thùng hiện tại gồm: 1 5 5

→ MAX: 5, MIN: 1 → Số tiền giải thưởng: $5 - 1 = 4$

Trong thùng còn: 5

Ngày cuối có 1 hóa đơn: 2

Thùng hiện tại gồm: 2 5

→ MAX: 5, MIN: 2 → Số tiền: $5 - 2 = 3$

Trong thùng còn:

➔ Tổng số tiền giải thưởng: $2 + 1 + 9 + 4 + 3 = 19$

Hướng dẫn giải

Ý tưởng cơ bản

Mỗi ngày phải lấy được MAX và MIN.

- Sử dụng 2 cấu trúc heap: max heap để lấy giá trị hóa đơn MAX, min heap để lấy giá trị hóa đơn MIN.

Nhưng không được lấy trùng?

- Đánh số thứ tự cho từng hóa đơn. Mỗi lần lấy một hóa đơn từ max heap hoặc min heap sẽ kiểm tra: nếu hóa đơn này đã được lấy ra trước đó thì bỏ qua và lấy hóa đơn kế tiếp. Sau khi đã lấy ra cũng phải đánh dấu lại rằng hóa đơn này đã được lấy ra.
- Khi bỏ hóa đơn vào heap, phải đi kèm với số thứ tự của hóa đơn đó.
- Sử dụng mảng taken để đánh dấu, `taken[i] = True` nghĩa là **hóa đơn thứ i** đã từng được lấy ra, và ngược lại.

Ví dụ ý tưởng

Input:

3

3 1 2 3

2 1

1 6 2

Khởi tạo mảng taken toàn bộ là False như sau:

Số thứ tự	0	1	2	3	4	5
Đánh dấu	F	F	F	F	F	F
(Giá trị hóa đơn tương ứng)	1	2	3	1	6	2

Output:

Ví dụ ý tưởng

Input:

3

3 1 2 3

2 1

1 6 2

Số thứ tự	0	1	2	3	4	5
Đánh dấu	F	F	F	F	F	F
(Giá trị hóa đơn tương ứng)	1	2	3	1	6	2

Ngày đầu tiên, thêm (1, 0), (2, 1) và (3, 2) vào heap:

→ Max heap: (3, 2), (2, 1), (1, 0)

→ Min heap: (1, 0), (2, 1), (3, 2)

Output:

Lấy MAX: 3 (số thứ tự 2), đánh dấu taken[2] = True

Lấy MIN: 1 (số thứ tự 0), đánh dấu taken[0] = True

Số tiền giải thưởng: 3 - 1 = 2

Ví dụ ý tưởng

Input:

3

3 1 2 3

2 1

1 6 2

Số thứ tự	0	1	2	3	4	5
Đánh dấu	T	F	T	F	F	F
(Giá trị hóa đơn tương ứng)	1	2	3	1	6	2

Ngày thứ hai, thêm (1, 3) vào heap:

→ Max heap: (2, 1), (1, 3), (1, 0)

→ Min heap: (1, 3), (2, 1), (3, 2)

Output:

Lấy MAX: 2 (số thứ tự 1), đánh dấu taken[1] = True

Lấy MIN: 1 (số thứ tự 3), đánh dấu taken[3] = True

Số tiền giải thưởng: 2 - 1 = 1.

Ví dụ ý tưởng

Input:

3

3 1 2 3

2 1

1 6 2

Số thứ tự	0	1	2	3	4	5
Đánh dấu	T	T	T	T	F	F
(Giá trị hóa đơn tương ứng)	1	2	3	1	6	2

Ngày thứ ba, thêm (6, 4), (2, 5) vào heap:

→ Max heap: (6, 4), (2, 5), (1, 3), (1, 0)

→ Min heap: (2, 1), (2, 5), (3, 2), (6, 4)

Output:

Lấy MAX: 6 (số thứ tự 4), đánh dấu taken[4] = True

Lấy MIN: 2 (số thứ tự 1), nhưng taken[1] == True

→ Hóa đơn này đã được lấy ra trước đó → Bỏ qua.

Ví dụ ý tưởng

Input:

3

3 1 2 3

2 1

1 6 2

Số thứ tự	0	1	2	3	4	5
Đánh dấu	T	T	T	T	T	F
(Giá trị hóa đơn tương ứng)	1	2	3	1	6	2

Ngày thứ ba, thêm (6, 4), (2, 5) vào heap:

→ Max heap: (6, 4), (2, 1), (1, 3), (1, 0)

→ Min heap: (2, 5), (3, 2), (6, 4)

Output:

7

Lấy MAX: 6 (số thứ tự 4), đánh dấu taken[4] = True

Lấy MIN: 2 (số thứ tự 5), đánh dấu taken[5] = True → OK

Số tiền giải thưởng: 6 - 2 = 4

→ Tổng tiền thưởng mà siêu thị phải chi trả là

= 2 + 1 + 4 = 7.

Các bước giải

Bước 1: Nhập dữ liệu.

Bước 2: Khai báo một maxHeap và một minHeap để lưu trữ các hóa đơn, mảng taken để đánh dấu.

Bước 3: Mỗi ngày, đưa các hóa đơn trong ngày vào maxHeap và minheap.

Bước 4: Lấy ra một hóa đơn chưa được sử dụng trong maxHeap và một hóa đơn chưa được sử dụng trong minHeap, tính chênh lệch và đánh dấu lại.

Bước 5: Đưa ra kết quả.

Độ phức tạp: $O(M \cdot \log M)$, Với M là tổng số lượng hóa đơn

Mã giả

Mã giả

```
Read(N)
maxHeap = []
minHeap = []
Index = 0
SumPrize = 0
taken = [False, False, False, ...]
for i in [1, N]:
    read(k)
    for j in [1, k]:
        read(u)
        maxHeap.push((u, index))
        minheap.push((u, index))
        index += 1
    # To be continue
```

Mã giả

```
while (taken[maxHeap.top().second]):  
    maxHeap.pop()  
while (taken[minHeap.top().second]):  
    minHeap.pop()  
SumPrize += maxHeap.top() - minHeap.top()  
taken[maxHeap.top().second] = True  
taken[minHeap.top().second] = True  
maxHeap.pop()  
minHeap.pop()  
Print(SumPrize)
```