

# Monk and Multiplication

Big-O Blue Online Lecture 07: Heap

# Tóm Tắt Đề Bài

# Tóm Tắt Đề Bài

Cho một mảng A gồm N phần tử, với mỗi index  $i$  ( $1 \leq i \leq N$ ), xác định tích của 3 số có giá trị lớn nhất trong đoạn  $[1, i]$ .

## Input:

Số nguyên N - số lượng phần tử trong mảng A:  $1 \leq N \leq 100000$ .

N số nguyên của mảng A:  $0 \leq A[i] \leq 1000000$

## Output:

N dòng, dòng thứ  $i$  là tích của 3 số có giá trị lớn nhất trong đoạn  $[1, i]$  của mảng. Nếu không tồn tại số lớn thứ hai hoặc số lớn thứ ba, xuất ra -1.

# Giải Thích Ví Dụ

# Giải Thích Ví Dụ

5

1 2 3 4 5

1	2	3	4	5
---	---	---	---	---

# Giải Thích Ví Dụ

5

1 2 3 4 5



Đoạn mảng xét hiện tại: {1}

Không có phần tử lớn thứ hai và lớn thứ 3

→ -1

# Giải Thích Ví Dụ

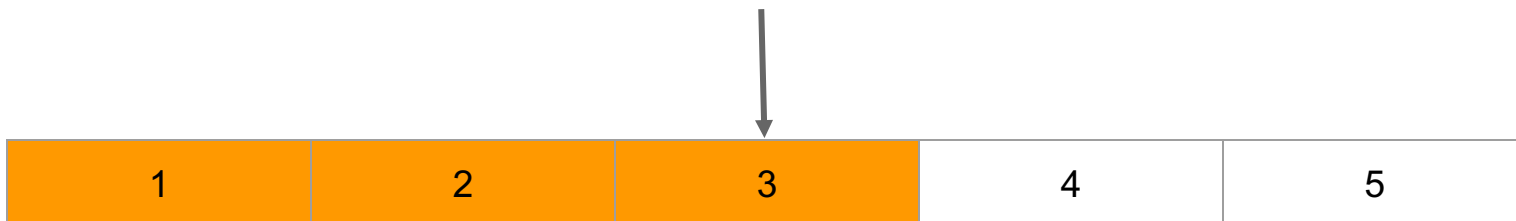


Đoạn mảng xét hiện tại: {1,2}

Không có phần tử lớn thứ ba

→ -1

# Giải Thích Ví Dụ



Đoạn mảng xét hiện tại: {1,2,3}

Ba số lớn nhất trong mảng: 3,2,1

$$\rightarrow 3*2*1 = 6$$



# Giải Thích Ví Dụ



Đoạn mảng xét hiện tại: {1,2,3,4}

Ba số lớn nhất trong mảng: 4,3,2

$$\rightarrow 4*3*2 = 24$$

# Giải Thích Ví Dụ



Đoạn mảng xét hiện tại: {1,2,3,4,5}

Ba số lớn nhất trong mảng: 5,4,3

$$\rightarrow 5 * 4 * 3 = 60$$

# Hướng Dẫn Giải

# Ý Tưởng

Cách 1: Lưu các phần tử hiện tại vào một list và sắp xếp list theo thứ tự (tăng dần hoặc giảm dần) để xác định 3 phần tử lớn nhất trong mảng hiện tại.

Cách 2: Dùng một cây Heap để lưu các phần tử trong mảng, xác định top 3 phần tử có giá trị lớn nhất.

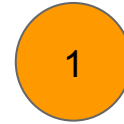
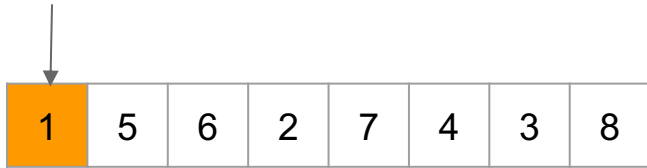
# Ý Tưởng

Input:

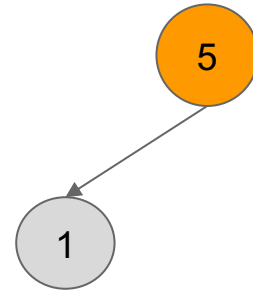
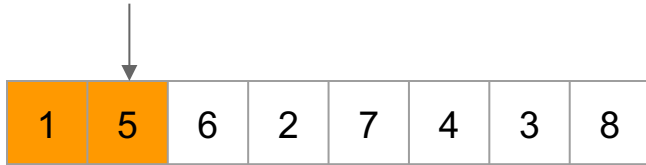
8

1 5 6 2 7 4 3 8

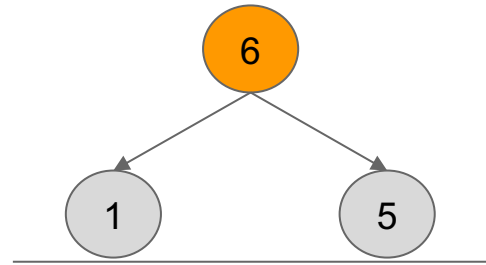
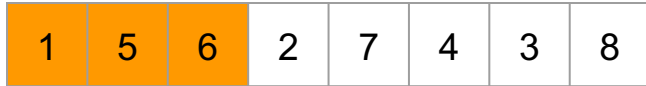
# Ý Tưởng



# Ý Tưởng

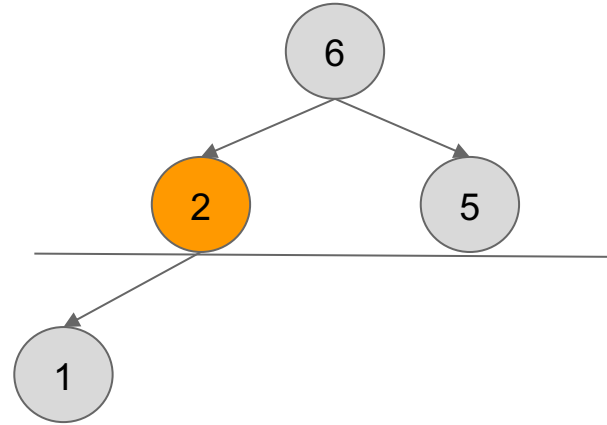
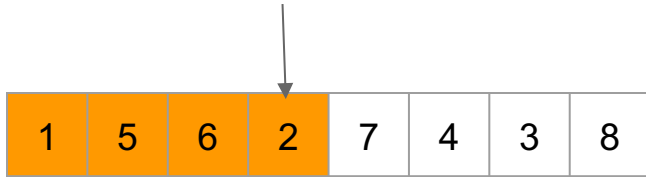


# Ý Tưởng

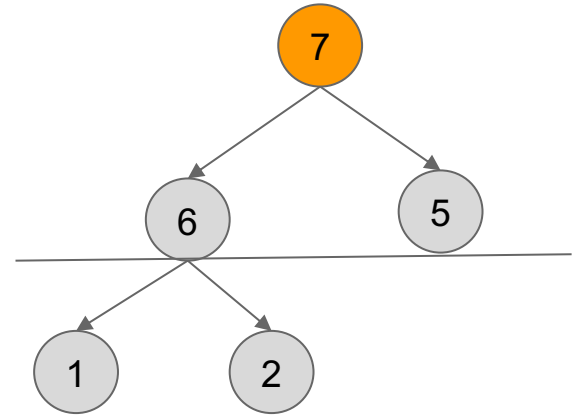
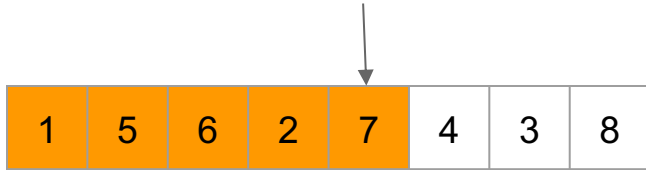




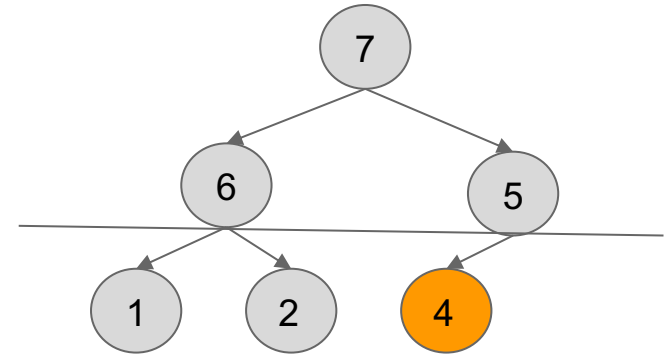
# Ý Tưởng



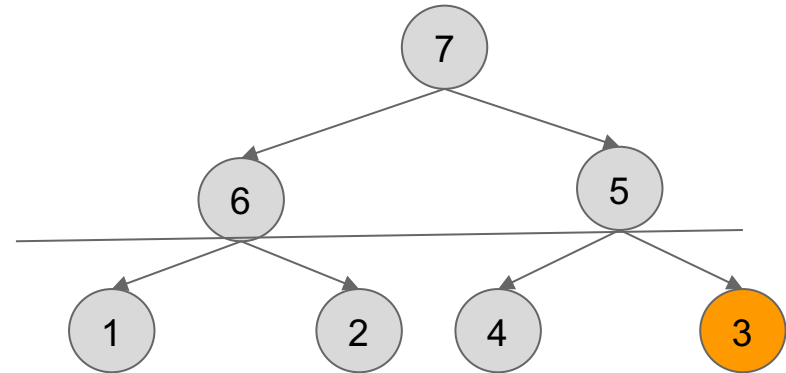
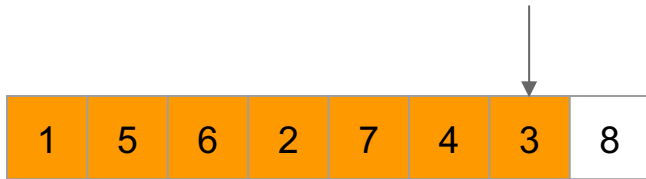
# Ý Tưởng



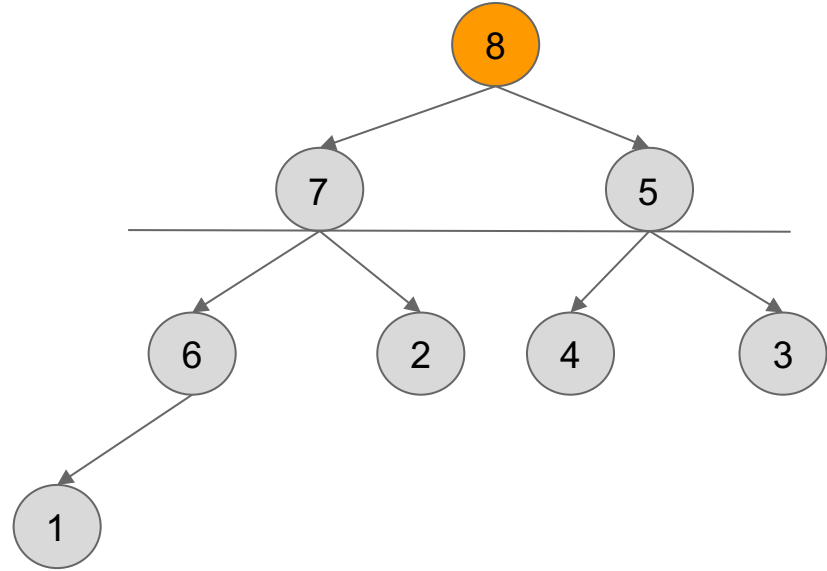
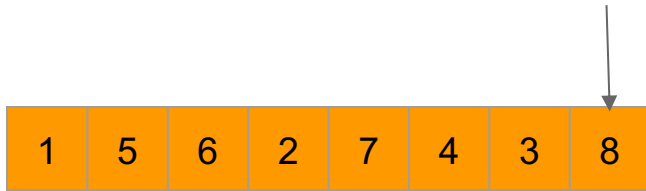
# Ý Tưởng



# Ý Tưởng



# Ý Tưởng



# Thuật Toán

Bước 1: Đọc input N

Bước 2: Khởi tạo 1 Max Heap, đọc mỗi phần tử của mảng và push vào Max Heap.

- Nếu kích thước Max Heap  $< 3$ :  $\rightarrow -1$
- Xác định phần tử lớn nhất của Max Heap. Pop phần tử lớn nhất đó ra khỏi Max Heap và tiếp tục xác định phần tử lớn nhất thứ hai. Tương tự để xác định phần tử lớn nhất thứ ba. Lưu các giá trị first, second, third này sau đó push lại vào Heap để duy trì mảng đang xét hiện tại.

Độ phức tạp:  $O(N \log N)$

# Mã Giả

# Mã Giả

```
read(N, A)

pq = PriorityQueue()

for i := 0 to N - 1 do:

    pq.push(A[i])

    if (pq.size() < 3):

        print(-1)

    else:

        a, b, c = pq.pop(), pq.pop(), pq.pop()

        print(a * b * c)

        pq.push(a, b, c)
```