



Monk and Multiplication

Link submit: <https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/monk-and-multiplication/>

Solution:

C++	http://ideone.com/sdCIXL
Java	http://ideone.com/8MWDMK
Python	http://ideone.com/kWVifX

Tóm tắt đề:

Monk đang tìm hiểu hàng đợi ưu tiên. Cho cậu một mảng số nguyên. Với mỗi phần tử khi thêm vào hàng đợi ưu tiên, Monk muốn tìm tích của ba phần tử lớn nhất hiện có trong hàng đợi.

Nếu không có đủ ba số lớn nhất, nhì và ba thì in ra -1.

Input:

Dòng đầu tiên chứa số N ($1 \leq N \leq 100.000$) là số lượng phần tử trong mảng.

Dòng tiếp theo chứa N phần tử trong mảng ($0 \leq A[i] \leq 1.000.000$).

Output:

Với mỗi phần tử khi thêm vào thì in ra tích của số lớn nhất, số lớn nhì, số lớn ba. Không có đủ ba số thì in ra -1.

Ví dụ:

5	-1
1 2 3 4 5	-1
	6
	24
	60

Giải thích ví dụ:

Thêm 1 vào hàng đợi ưu tiên \rightarrow in ra -1.

Thêm 2 vào hàng đợi ưu tiên \rightarrow in ra -1.

Thêm 3 vào hàng đợi ưu tiên \rightarrow in ra $3 * 2 * 1 = 6$.

Thêm 4 vào hàng đợi ưu tiên \rightarrow in ra $4 * 3 * 2 = 24$.

Thêm 5 vào hàng đợi ưu tiên \rightarrow in ra $5 * 4 * 3 = 60$.

Hướng dẫn giải:

Ta chỉ cần áp dụng hàng đợi ưu tiên là giải quyết được bài này.

Lưu các phần tử để cho vào một mảng.

Lần lượt duyệt qua các phần tử và bỏ vào hàng đợi ưu tiên. Mỗi lần bỏ phần tử mới vào thì lấy top 3 để nhân với nhau và in ra kết quả. Nếu như số lượng phần tử trong hàng đợi ưu tiên nhỏ hơn 3 thì ta xuất ra -1.

Độ phức tạp: $O(N \log N)$ với N là số lượng phần tử trong mảng.

Big-O Coding