



QHEAP1

Link: <https://www.hackerrank.com/challenges/qheap1/problem>

Solution:

C++	https://ideone.com/DBY3yA
Java	https://ideone.com/j7KikM
Python	https://ideone.com/4qovdg

Tóm tắt đề: Cho bạn các loại truy vấn như sau:

"1 v": Thêm một phần tử v vào heap.

"2 v": Xóa một phần tử khỏi heap.

"3": In phần tử nhỏ nhất trong heap.

Chú ý: Các phần tử cần xóa luôn luôn có trong heap và các phần tử trong heap là phân biệt nhau.

Input:

Dòng đầu tiên chứa số Q là số lượng Q cần thực hiện. $1 \leq Q \leq 10^5$.

Q dòng tiếp theo chứa một trong 3 loại truy vấn trên. $10^9 \leq v \leq 10^9$.

Output

Với truy vấn loại 3 in phần tử nhỏ nhất trong heap, mỗi phần tử trên 1 dòng.

5	4
1 4	9
1 9	
3	
2 4	
3	

Giải thích:

Thêm 4 vào hàng đợi ưu tiên → 4.

Thêm 9 vào hàng đợi ưu tiên → 4, 9.

In phần tử nhỏ nhất ra → in ra 4.

Xóa phần tử 4 khỏi heap \rightarrow 9.

In phần tử nhỏ nhất ra \rightarrow in ra 9.

Hướng dẫn giải:

Nhận xét: Trong 3 query thì query “delete” là cần lưu ý nhất. Vì để xóa 1 phần tử bất kỳ trong hàng đợi ưu tiên chúng ta sẽ phải pop toàn bộ các giá trị từ đầu cho đến khi nào gặp phần tử cần xóa thì ta sẽ ngưng.

Lúc này chúng ta sẽ phải lưu lại toàn bộ giá trị đã xóa lúc này ra để push lại vào trong hàng đợi ưu tiên.

Ví dụ hàng đợi ưu tiên đang có: 3 8 9 12 15. Chúng ta muốn xóa giá trị 9 thì phải đem 3 và 8 ra ngoài sau đó xóa 9 đi rồi bỏ lại 3 và 8 vào hàng đợi ưu tiên. Khi thao tác như vậy, nếu phần tử cần xóa nằm ở vị trí cuối cùng của hàng đợi ưu tiên vậy ta phải sử dụng 2 vòng lặp. 1 vòng lặp để xóa và một vòng lặp để bỏ lại.

\Rightarrow Không khả thi vì số lượng phần tử 10^5 .

Cải tiến: Để việc xóa không trở nên phức tạp bạn sẽ dùng thêm một hàng đợi ưu tiên khác (tạm đặt tên là pq_remove. Khi gặp một query “delete” thay vì xóa bạn sẽ thêm phần tử cần xóa này vào hàng đợi pq_remove.

Lúc này ở thao tác in phần tử ra màn hình (thao tác số 3) mỗi lần in bạn cần phải kiểm tra xem phần tử ở hàng đợi ban đầu có trùng với phần tử ở hàng đợi pq_remove không? Nếu trùng nhau nghĩa là phần tử này trước sẽ bị xóa đi. Lúc này bạn mới xóa phần tử này đi trong hàng đợi ban đầu. Còn nếu không trùng nhau thì bạn vẫn in ra giá trị ra bình thường.

Ví dụ: hàng đợi ban đầu 3 8 9 12 15. Bạn có một truy vấn xóa phần tử số 9. Thay vì xóa trong hàng đợi ban đầu bạn thêm số 9 vào pq_remove: 9.

Khi đề bài có thao tác in giá trị phần tử ra thì bạn kiểm tra pq.top có trùng với pq_remove.top hay không, nếu không trùng thì in số 3 ra. Còn nếu trùng thì đơn giản là chỉ việc xóa phần tử 9 ra khỏi pq và pq_remove.

(Bạn có thể xem source code để hiểu thêm ý tưởng này).

Độ phức tạp sau khi cải tiến: $O(Q \log N)$ với Q là số lượng truy vấn, N là số lượng phần tử được add vào hàng đợi ưu tiên.