

## <BOJ 1927\_최소 힙>

### 문제

최소 힙을 이용하여 다음과 같은 연산을 지원하는 프로그램을 작성

1. 배열에 자연수  $x$ 를 넣는다.
2. 배열에서 가장 작은 값을 출력하고, 그 값을 배열에서 제거한다

### 입력

첫째 줄: 연산의 개수  $N$  ( $1 \leq N \leq 100,000$ )

다음  $N$ 개의 줄:

자연수  $x$  : 배열(또는 힙)에  $x$ 를 추가하는 연산

$x = 0$  : 배열에서 가장 작은 값을 출력하고 제거하는 연산

(단, 배열이 비어 있으면 0 출력)

※  $x$ 는 0 또는  $2^{31}$ 보다 작은 자연수만 입력됨 (음수 없음)

### 출력

입력에서 0이 주어진 횟수만큼 답을 출력, 만약 배열이 비어 있는 경우인데 가장 작은 값을 출력하라고 한 경우에는 0을 출력

### 예제 입력 1

```
9
0
12345678
1
2
0
0
0
0
0
32
```

### 예제 출력 1

```
0
1
2
12345678
0
```

#### 접근 방법

1. push() 함수를 통해 새로운 값을 힙의 맨 끝에 추가한 뒤, 부모 노드와 비교하여 부모 노드보다 작으면 서로 위치를 바꾼다. 이 과정을 반복하여 힙의 성질을 만족할 때까지 위로 올라가며 수행한다.
2. top() 함수는 힙의 최상단에 있는 최소값(heap[1])을 반환하며, 힙이 비어 있으면 0을 반환한다.
3. pop() 함수는 루트 노드(최소값)와 마지막 원소의 위치를 바꾼 뒤, 힙 크기를 줄여 마지막 원소를 제거한다. 이후 자식 노드와 비교하면서, 왼쪽 자식과 오른쪽 자식 중 더 작은 자식과 위치를 바꾸며 힙의 성질을 만족할 때까지 아래로 내려가며 수행한다.

```

/*
1927/최소 힙/BOJ
12ms
32m
*/
#include <bits/stdc++.h>
using namespace std;

int heap[100005];
int sz = 0;

void push(int x){
    sz++;
    heap[sz] = x;
    int i = sz;
    while(i > 1 && heap[i/2] > heap[i]){
        swap(heap[i], heap[i/2]);
        i = i / 2;
    }
}

int top(){
    if(sz == 0) return 0;
    return heap[1];
}

void pop(){
    if(sz == 0) return;

    heap[1] = heap[sz];
    sz--;

    int i = 1;
    while(i * 2 <= sz){
        int left = i * 2;
        int right = i * 2 + 1;
        int minch = left;

        if(right <= sz && heap[right] < heap[left])
            minch = right;
    }
}

```

```

        if(heap[i] > heap[minch]){
            swap(heap[i], heap[minch]);
            i = minch;
        } else {
            break;
        }
    }
}

int main(void){
    ios::sync_with_stdio(0);
    cin.tie(0);

    int N, n;
    cin >> N;
    for(int i = 0; i < N; i++){
        cin >> n;
        if(n == 0){
            cout << top() << '\n';
            pop();
        } else {
            push(n);
        }
    }
}

```

```

//STL 사용
#include<bits/stdc++.h>

using namespace std;

int main(){

    ios::sync_with_stdio(0);
    cin.tie(0);
    priority_queue<int, vector<int>, greater<int>> pq;

    int N;
    unsigned int m;

    cin>>N;
    for(int i=0;i<N;i++){
        cin>>m;
        if(m==0){
            if(pq.empty()){
                cout<<0<<"\n";
            }
            else{
                cout<<pq.top()<<"\n";
                pq.pop();
            }
        }
        else{
            pq.push(m);
        }
    }
}

```