

# Heap Sort

주어진 숫자를 오름차순으로 정렬하는 힙 정렬은 아래와 같이 두 단계로 이루어 진다.

(1) 최대 힙 구성(max heap construction)

(2) 단계 (1)에서 구성된 최대 힙에서 루트 노드에 있는 최대값을 제거한다. 이때 힙을 재구성하여야 한다. 제거된 최대값은 힙을 재구성 하기 전의 힙의 마지막 원소가 저장된 배열의 원소에 저장한다. 이 과정을 반복하여 힙에 속한 모든 정수가 제거될 때까지 실행한다.

이런 과정을 의사코드(슈도코드, pseudo-code)로 작성하면 아래와 같다.

## Algorithm 2.10 Heapsort

*Input:*  $L$ , an unsorted array, and  $n \geq 1$ , the number of keys.

*Output:*  $L$ , with keys in nondecreasing order.

```
procedure Heapsort (var  $L$ : Array;  $n$ : integer);  
var  
     $i$ ,  $heapsize$  : Index;  
     $max$  : Key;  
begin  
    { Heap Construction }  
    for  $i := \lfloor n/2 \rfloor$  to 1 by -1 do  
        FixHeap ( $i$ ,  $L[i]$ ,  $n$ )  
    end { for };  
    { Repeatedly remove the key at the root and rearrange the heap. }  
    for  $heapsize := n$  to 2 by -1 do  
         $max := L[1]$ ;  
        FixHeap (1,  $L[heapsize]$ ,  $heapsize-1$ );  
         $L[heapsize] := max$   
    end { for }  
end { Heapsort }
```

## Algorithm 2.8 FixHeap

*Input:* The root of a heap and a key  $K$  to be inserted.

*Output:* The heap with keys properly rearranged.

```
procedure FixHeap ( $root$ : Node;  $K$ : Key);  
var  
     $vacant$ ,  $largerChild$  : Node;  
begin  
     $vacant := root$ ;  
    while  $vacant$  is not a leaf do  
         $largerChild :=$  the child of  $vacant$  with the larger key;  
        if  $K < largerChild$ 's key  
            then  
                copy  $largerChild$ 's key to  $vacant$ ;  
                 $vacant := largerChild$   
            else exitloop  
        end { if }  
    end { while };  
    put  $K$  in  $vacant$   
end { FixHeap }
```

기본연산자: 비교연산자  
(비교연산 횟수: 최대 2 번)  
왼쪽 child 만 있는 경우 : 1 번  
child가 두 개 있는 경우 : 2 번

<참고자료: Foundations of Algorithms Using C++ Pseudocode (Second Edition), Neapolitan, Naimipour, 1998>

주어진 숫자를 힙 정렬하는 과정에서 위 의사코드에서 실행되는 비교연산자의 횟수를 계산하는 프로그램을 작성하시오.

## 입력

입력은 표준입력(standard input)을 사용한다. 입력은  $t$  개의 테스트 케이스로 주어진다. 입력 파일의 첫 번째 줄에 테스트 케이스의 개수를 나타내는 정수  $t$  가 주어진다. 두 번째 줄부터  $t$  개의 줄에는 한 줄에 한 개의 테스트 케이스에 해당하는 정수들이 주어진다. 각 테스트 케이스에 해당되는 각 줄의 첫 번째 정수  $n$  ( $1 \leq n \leq 1,000$ ) 은 정렬하여야 할 자연수의 개수를 나타낸다. 그 다음에는  $n$  개의 자연수가 주어진다. 같은 줄에 나열되는 각 숫자들 사이에는 한 개의 공백이 있으며, 잘못된 데이터가 입력되는 경우는 없다.

## 출력

출력은 표준출력(standard output)을 사용한다. 입력되는 테스트 케이스의 순서대로 다음 줄에 이어서 각 테스트 케이스의 결과를 출력한다. 각 테스트 케이스에 해당하는 출력의 첫 줄에 입력되는  $n$  개의 정수를 문제에서 제시된 의사코드로 구현할 때, 기본 연산자의 실행 횟수를 출력한다.

## 입력과 출력의 예

입력	출력
4	77
15 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	66
15 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1	41
10 7 3 2 9 5 1 10 8 4 6	0
1 1	