

〈2021-2. 알고리즘 및 실습〉 - 과제1(OJ)

※ 입출력에 대한 안내

- 특별한 언급이 없으면 문제의 조건에 맞지 않는 입력은 입력되지 않는다고 가정하라.
- 입출력 예시에서 \mapsto 이 후는 각 입력과 출력에 대한 설명이다.
- 소스 코드 체크 후 감점이 될 수 있음. (정해진 규칙을 지키지 않으면 감점)

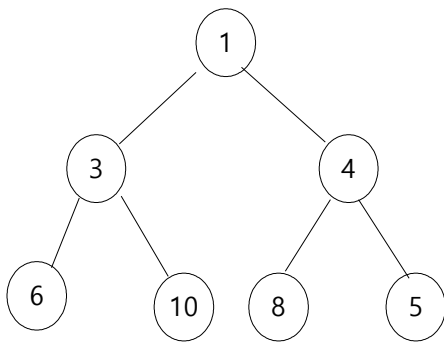
[문제 1] (50) 주어진 숫자들이 나타내는 heap이 min-heap인지 아닌지 체크하는 프로그램을 작성하여라. min-heap이면 Yes, min-heap이 아니면 No를 출력하고, 마지막 내부노드의 값을 출력한다. 편의상 중복숫자는 없다고 가정한다. 불필요한 비교를 남발하면 감점. 내부노드가 하나도 없는 경우는 없다.

입력 예시 1

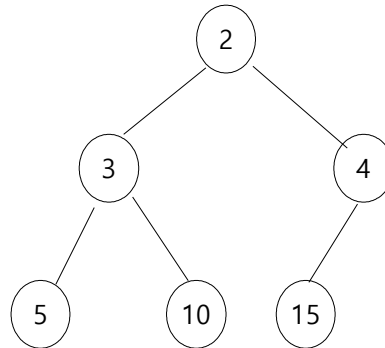
7 \mapsto N : heap의 노드 수 1 3 4 6 10 8 5
--

출력 예시 1

Yes 4 \leftarrow 마지막 내부노드의 값



예시 1



예시 2

입력 예시 2

6 \mapsto N : heap의 노드 수 2 3 4 5 10 15

출력 예시 2

Yes 4

[문제 2] (50) 양의 정수 N과 N개의 정수가 입력이 된다. (K개의 노드만 갖는 힙을 사용하여 N개 중 K개 큰 값들을 출력하려고 한다. $K \leq N$ 힙을 위하여 (K+1)개 힙노드만 동적 할당하여야 한다. 힙을 사용하여 작업을 하면 정렬 되지 않은 K개의 큰 값들이 결과로 나오는데, OJ에서의 채점을 위하여 오름차순으로 정렬하여 출력하여라. (중복은 없다고 가정한다.)

입력 예시 1

10 5 \mapsto N : 입력 수, K : heap의 노드 수
1 3 4 6 10 8 5 0 2 7

출력 예시 1

5 6 7 8 10

입력 예시 2

5 2 \mapsto N : 입력 수, K : heap의 노드 수
4 10 3 5 1

출력 예시 2

5 10

- 1) 입력은 N개중 K개만 먼저 입력이 되어 있다고 가정하고 BuildHeap
- 2) 이후에 하나씩 더 추가로 입력이 들어온다고 가정하고 작업을 할 것.
- 3) 효율적인 방법을 사용할 것.(BuildHeap & 나머지작업)

입력예시1의 경우 아래와 같은 heap 트리 내에서 작업을 해야 함.

