

# Week 10

- 주의 사항: 부정행위 금지(채점서버 외 인터넷 사용금지), STL 사용금지 (string, vector는 사용 가능), **힙 기반**의 우선순위 큐를 구현하여 문제를 해결할 것.
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

## 문제 1

자연수  $N$  ( $1 \leq N \leq 10,000$ )를 입력 받아 힙(heap)을 기반으로 우선순위 큐를 생성하는 프로그램을 만들어보자. 힙은 **최대 힙(Max-Heap)**으로 구성되며, 힙에 입력되는 엔트리의 키( $E_{key}$ )와 값( $E_{value}$ )은 동일한 값이다. 힙을 생성한 후, 명령어를 입력 받고 명령어에 따른 기능을 실행한다. 이 때 입력 받을 명령어가 다음과 같이 주어진다. (단, 힙에 입력되는 엔트리의 키( $E_{key}$ )가 중복되는 경우는 존재하지 않는다.)

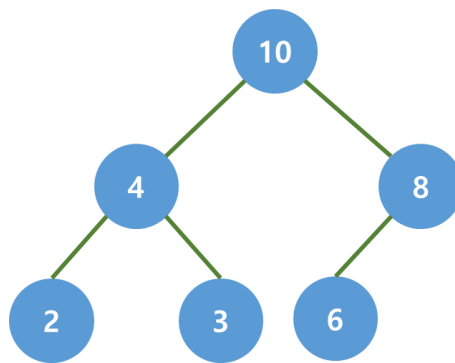


그림 1. [8, 4, 6, 2, 3, 10]이 차례대로 최대 힙에 입력된 예

- **insert( $i$ )**: 자연수  $i$  ( $1 \leq i \leq 10,000$ )를 우선순위 큐에 삽입한다. (즉,  $E_{key} = E_{value} = i$ )
- **size()**: 힙에 저장 되어 있는 엔트리의 개수를 출력한다.
- **isEmpty()**: 힙이 비어 있으면 1을, 비어 있지 않다면 0을 출력한다.
- **pop()**: 우선순위가 가장 높은 엔트리의 값을 출력 후 해당 엔트리를 제거한다. 만약 힙이 비어 있는 경우에는, -1을 출력한다.
- **top()**: 우선순위가 가장 높은 엔트리의 값을 출력한다. 만약 힙이 비어 있는 경우에는, -1을 출력한다.
- **print()**: 힙의 원소를 위부터 아래로, 왼쪽에서 오른쪽으로 차례대로 출력한다. 단, 비어있는 경우 -1을 출력한다. (위의 그림 1을 예로서 든다면, [10, 4, 8, 2, 3, 6]이 출력된다)

# 입력

첫 번째 줄에 테스트 케이스 수  $T(1 \leq T \leq 1,000)$ 가 주어진다. 두 번째 줄부터  $T$ 개의 줄에 명령어가 하나씩 주어진다.

# 출력

출력해야 하는 명령어가 주어질 때마다 그 결과를 한 줄씩 출력한다.

## 예제 입출력

예제 입력	예제 출력
17 isEmpty print insert 10 insert 5 size insert 6 insert 9 pop insert 2 insert 4 top insert 8 insert 11 pop insert 15 insert 3 print	1 -1 2 10 9 11 15 5 9 3 4 6 8 2