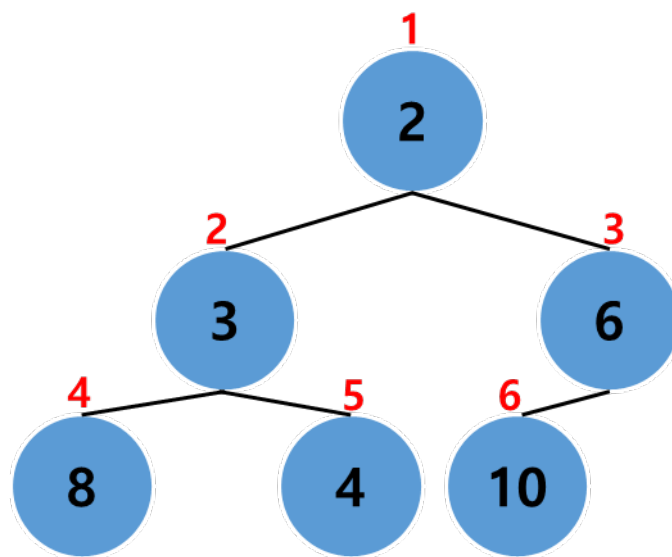


# Week 10

- 주의 사항: 부정행위 금지(채점서버 외 인터넷 사용금지), STL 사용금지(string, vector는 사용 가능), **힙 기반**의 우선순위 큐를 구현하여 문제를 해결할것.
- 힙(heap) 생성시, upHeap을 이용하여 원소를 하나씩 힙에 삽입하는 방식으로 구현할 것.
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

## 문제 2

양의 정수를 입력 받아 배열(array) 기반의 힙(Heap)을 생성하여 입력된 숫자를 정렬하는 프로그램을 만들어보자.



예를 들어, 숫자가 8 4 6 2 3 10으로 입력이 되었을 때, 최소힙(Min-Heap)은 위의 위 그림과 같은 형태로 생성이 된다.

위 최소힙을 인덱스(붉은색 숫자) 순으로 읽으면 2 3 6 8 4 10이 되며, 숫자를 오름차순으로 정렬한 결과는 2 3 4 6 8 10이다. 그리고 위 최소힙을 생성하기 위해 시행되는 upheap의 횟수는 10회다.

(단, 힙에 입력되는 key 값 중 중복되는 값은 존재하지 않는다.)

## 입력

1. 첫 번째 줄에 테스트 케이스의 개수  $T$ 가 주어진다. ( $1 \leq T \leq 20$ ).
2. 각 케이스마다 다음이 반복되어 나타난다.
  - 1) 각 케이스의 첫 번째 줄에 입력 받을 숫자의 개수  $N$ 와, 탐색해야 할 노드의 인덱스  $p$ 가 공백을 기준으로 주어진다. ( $1 \leq p \leq N \leq 10,000$ )
  - 2) 각 케이스의 두 번째 줄부터  $N$ 개의 숫자가 입력된다. 입력되는 숫자는 1~200,000으로 표현 가능한 양의 정수이다. 단, 입력되는 숫자에 중복되는 수는 존재하지 않는다.

# 출력

각 케이스 별로 아래와 같은 양식으로 출력한다.

- 1. 첫 번째 줄에는 주어진 정수를 기반으로 생성된 최소힙을 공백을 기준으로 출력한다.
- 2. 두 번째 줄에는 생성된 최소힙의 p번째 인덱스 값을 출력한다.
- 3. 세 번째 줄에는 주어진 정수를 오름차순으로 정렬한 결과에서 p번째 값을 출력한다.
- 4. 네 번째 줄에는 최소힙을 생성할 때 호출 된 upheap의 횟수를 출력한다.

## 예제 입출력

예제 입력	예제 출력	// 는 참고용
1	2 3 6 8 4 10	
6 4	8	// 2 3 6 8 4 10
8 4 6 2 3 10	6	// 2 3 4 6 8 10
	10	