

# 과제1. Heap

---

## 과제 목적

Heap 자료구조의 삽입 삭제에 대한 연산 이해

Functor의 사용을 구현하여 comparator의 개념을 이해

## 과제 내용

### 1. Heap의 동작 이해 (10점)

heap에 원소를 삽입, 삭제할 때 heap의 구조 변화를 트리 형태로 그린다.

heap은 최소힙으로 동작하고 입력되는 원소는 양수만 가진다.

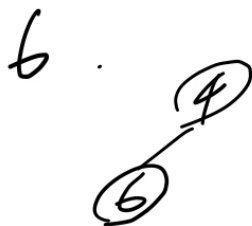
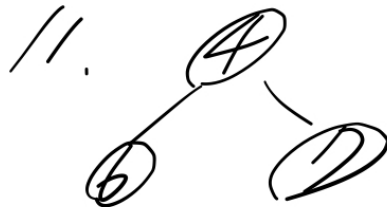
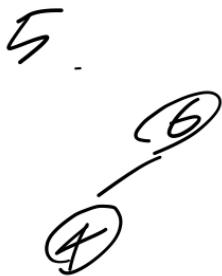
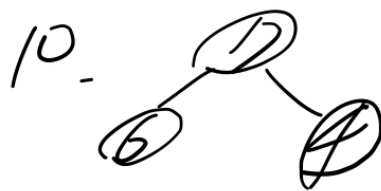
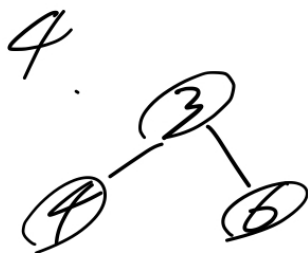
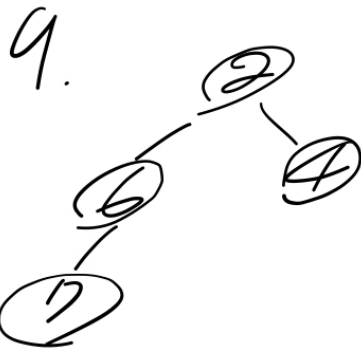
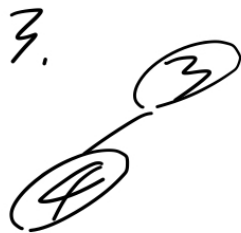
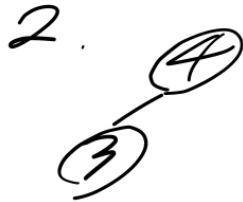
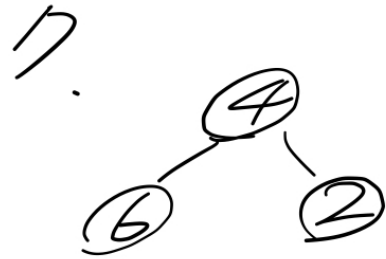
input에서 양수는 해당 정수를 heap에 삽입하고, -1은 heap에서 원소를 삭제함을 의미한다.

heap의 구조 변화는 트리에서 변화가 발생할 때 마다 그림을 그려 표현한다.

다음과 같은 input에 대해서 아래와 같은 그림을 그린다.



input : 4 3 6 -1 2 7 -1



입력은 총 3개의 케이스가 주어지고, 각 케이스를 위의 그림과 같이 트리 구조 변화를 그림으로 보여라.



input1 : 3 7 2 6 -1 4 5



input2 : 9 7 4 6 3 -1 -1 2 5 -1



input3 : 2 5 1 3 4 7 6 -1 -1 -1 3 2 1

## 2. Heap의 구현 (15점)

주어진 heap.cpp 파일에서 heap을 구현한다.

heap의 우선순위는 최소 heap을 만족해야 한다.

코드 작성은 인터페이스로 주어진 heap.cpp에서 작성한다.

heap.cpp에서 `insert` 함수와 `erase` 함수를 제외한 나머지 모든 함수는 변경하지 않는다.

정의된 함수에서 추가적인 인자는 허용하지 않는다.

별도의 함수가 필요시 추가적인 함수 정의는 허용한다.

코드를 작성하는 과정에서 STL의 사용은 허용하지 않는다.

`print_heap` 함수에서 출력되는 결과는 1번에서 작성한 그림과 일치하게 출력 되어야 한다.

## 3. Priority Queue에서 Comparator 구현 (0.1점)

주어진 priority\_queue.cpp 파일에서 priority queue를 구현한다.

priority queue의 우선순위는 클래스를 선언할 때 전달하는 comparator에 따라 달라진다.

정수 타입을 사용하는 priority queue는 comparator는 STL에 정의된 `less` 와 `greater` 를 사용한다.

`less` 의 경우 큰 수가 우선순위가 높고, `greater` 의 경우 작은 수가 우선순위가 높다.

`Student` 클래스의 우선순위는 `Comparator` 클래스에서 함수 객체를 직접 정의한다.

학생 객체를 사용하는 priority queue의 경우 학번이 낮은 경우가 우선순위가 높다.

별도의 클래스나 함수의 추가는 허용하지 않는다.

필요시 `Student` 클래스의 추가적인 멤버 함수의 정의는 허용한다.

코드를 작성하는 과정에서 STL의 사용은 허용하지 않는다.

## 과제 제출

과제 제출 마감 : 5월 22일 23시 59분

제출 방법 : 과제의 결과물들을 압축하여 튜터에게 개인 카톡으로 제출

- 1번 과제 : 작성한 그림을 pdf로 변환하여 제출
- 2번 과제 : 작성한 heap.cpp 파일 제출
- 3번 과제 : 작성한 priority\_queue.cpp 파일 제출

## 유의사항

**마감일이 얼마 남지 않은 수업 과제가 있는 경우 해당 과제는 하지 말것**

과제 내용이 어렵더라도 1번 과제는 꼭 해보길 바람

인터페이스로 주어진 코드에서 TODO 외의 부분은 작성하지 말것

과제 점수가 0.1점인 의미는 하기 싫다면 안해도 된다는 의미임