

## 자료 구조 Lab 011 :

lab011.zip 파일 : LabTest.java lab011.java lab.in lab.out lab011.pdf

-----

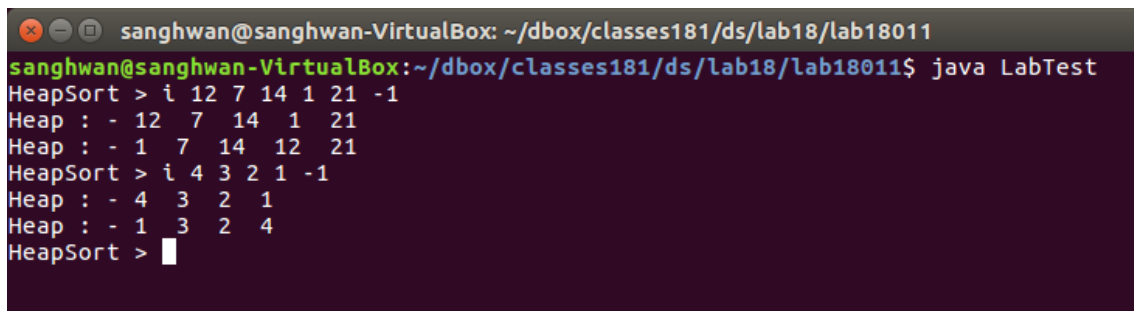
### 제출

lab011.java 를 학번.java 로 변경하여 이 파일 한 개만 제출할 것.

---

이번 숙제는 Heap Sort 알고리즘에서 Heapify의 기능을 구현하는 내용이다. **내림차순으로** 정렬하기 위한 heap을 구성하면 된다. lab011.java에는 HeapSort라는 클래스가 구현되어 있다. 사용자로부터 숫자를 입력 받고, 이 숫자들을 배열에 입력한 후 다시 Heap으로 구성한다.

수행 예는 다음과 같다.



```
sanghwan@sanghwan-VirtualBox: ~/dbox/classes181/ds/lab18/lab18011
sanghwan@sanghwan-VirtualBox:~/dbox/classes181/ds/lab18/lab18011$ java LabTest
HeapSort > i 12 7 14 1 21 -1
Heap : - 12 7 14 1 21
Heap : - 1 7 14 12 21
HeapSort > i 4 3 2 1 -1
Heap : - 4 3 2 1
Heap : - 1 3 2 4
HeapSort > 
```

사용자가 사용하는 명령어의 syntax는 다음과 같다. main() 함수에 정의되어 있다.

- i heapvalues

정렬할 숫자들을 입력받는 과정을 수행한다. 주어진 정수 값들을 -1이 나올 때까지 읽어서 그 내용대로 heap의 1번 index부터 차례로 집어 넣는다. 숫자들은 임의의 순서로 들어온다. 그리고 사용자로부터 입력 받은 숫자를 Heap으로 구성한다.

클래스 HeapSort에는 heapArr이라는 멤버 변수가 있는데 이 변수는 array 타입으로 사용자로부터 입력 받은 숫자를 저장한다. 그 숫자의 개수는 hSize 에 저장되어 있고, 숫자들은 배열의 인덱스 1부터 hSize 까지 차례대로 저장된다. 배열 인덱스 0은 사용하지 않는다. Heapify를 구현하기 위해 다음 2 함수를 구현해야 한다.

- void Adjust(int root, int n);

인덱스 root에서부터 시작하는 subtree를 min heap 상태를 만족하도록 변경한다.

이 heap의 맨 마지막 노드의 인덱스는 변수  $n$ 에 저장되어 있다.

- `void Heapify ();`

우선 `heapArr` 배열을 min heap의 상태로 변경한다. 즉 Heap Sort의 초기화 과정을 수행한다. 이를 위해 위에서 구현한 **Adjust** 함수를 적절하게 호출해야 한다. 알고리즘은 책이나 강의 자료를 참조한다.

## 프로그램 테스트

```
$ diff aaa lab.out
```

또는

```
$ diff -i --strip-trailing-cr -w aaa lab.out
```