

자료 구조 Lab 005 :

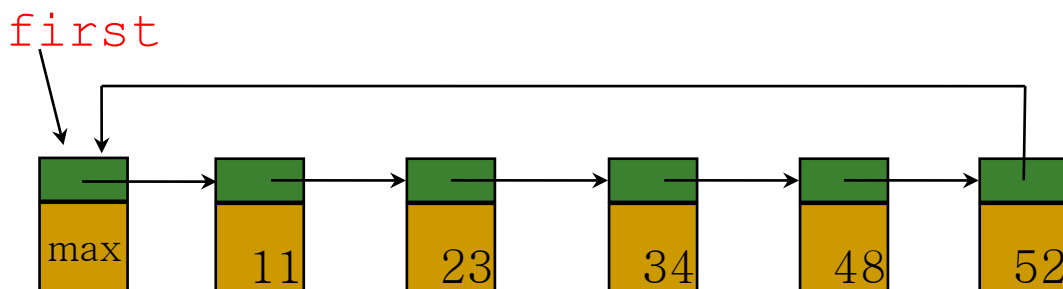
Lab17005.zip : LabTest.java, lab005.java, lab.in, lab.out, lab005.pdf

제출

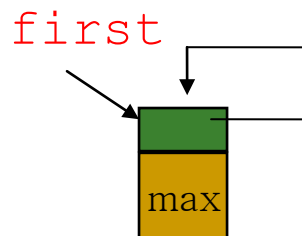
lab005.java 를 학번.java 로 변경하여 이 파일 한 개만 제출할 것.

다음은 Circular List를 이용하여 SortedList 라는 클래스를 구현하는 내용이다. SortedList 는 Circular List의 일종인데 데이터가 오름차순으로 정렬이 되어 있다.

SortedList클래스는 원소들을 아래와 같은 구조로 유지한다.



리스트가 비어 있을 경우는 다음과 같은 구조를 가진다.



단 헤더 노드에는 가장 큰 수가 저장되어 있다. 이는 구현의 편리성을 위해 저장한 값으로 실제 데이터와는 아무 상관 없다.

수행 예는 다음과 같다.

```
kmucs@localhost: ~/dbox/classes171/ds/lab17/lab17005
kmucs@localhost:~/dbox/classes171/ds/lab17/lab17005$ java LabTest
SL >
put 10
List : 10
Av : 0
SL >
put 20
List : 10 20
Av : 0
SL >
put 30
List : 10 20 30
Av : 0
SL >
put 40
List : 10 20 30 40
Av : 0
SL >
clear
List :
Av : 4
SL >
put 7
List : 7
Av : 3
SL >
put 9
List : 7 9
Av : 2
SL >
```

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

- put integervalue

정수인 integervalue 값을 가지는 노드를 정렬이 되는 위치를 찾아 삽입한다.

- clear

리스트를 모두 av 로 return 한다.

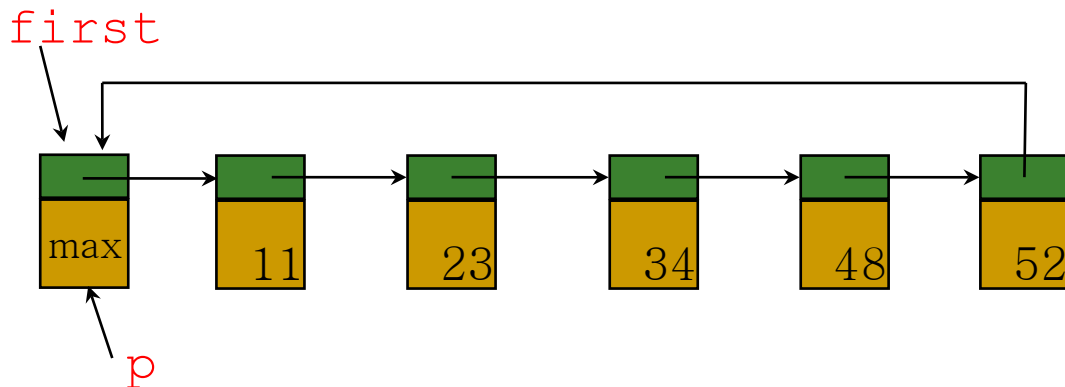
위에서 List : 는 현재 SortedList에 있는 원소를 보여준다. Av : 는 av 변수가 가리키고 있는 노드의 수를 보여준다.

이 내용을 구현하기 위해 다음 세 함수를 구현해야 한다.

- void Insert(T e);

데이터 값이 e 인 SortedListNode<T> 타입의 노드를 생성하여, 정렬을 만족하는 위치에 삽입한다. 중요한 점은 헤더 노드의 데이터 값이 max integer라는 것을 고려하여 구현하면 위치를 찾는 데 좀 더 편리하다. 만약 같은 값을 가지는 노드가 존재해도 그냥 정렬이 되도록 삽입한다.

삽입할 위치를 찾는 것은 결국 삽입할 노드의 바로 앞에 존재할 노드를 찾는 것이다. 변수 p를 first부터 시작해서, p.link.data 값이 주어진 e 값보다 큰 맨 처음의 p를 찾아내면 된다. 참고로 값을 비교하기 위해서는 compareTo() 메소드를 사용하면 된다. 이 p가 바로 삽입될 노드의 바로 앞 노드를 가리키게 된다. 아래의 그림과 같다. 새로운 노드를 생성할 때는 new를 사용하는 대신에 GetNode() 함수를 사용해야 한다.



- void Clear();

first가 가리키고 있는 Circular List를 av 로 반환하는 작업을 한다. Init() 함수로 다시 first에 헤더 노드를 추가하기 때문에 first가 가리키는 모든 노드를 av 로 반환해야 한다.

- SortedListNode<T> GetNode();

av에 달린 노드를 하나 떼어 return 한다. SortedList constructor를 비롯하여 노드를 생성하는 모든 곳에서 GetNode()를 사용한다. 교과서의 GetNode 함수를 참조할 것.

주어진 .cpp 파일을 컴파일 하면 수행은 가능하지만 아직 구현이 안된 부분은 "NEED TO IMPLEMENT" 라는 메시지가 출력된다.

프로그램 테스트

컴파일

```
$ javac lab005.java LabTest.java
```

```
// warning 메시지는 무시해도 됨.
```

실행

```
$ java LabTest
```

주어진 **input**으로 실행

```
$ java LabTest < lab.in
```

주어진 **output**과 비교

```
$ java LabTest < lab.in > abc
```

```
$ diff abc lab.out
```

또는

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