

# Homework3 Technical Report

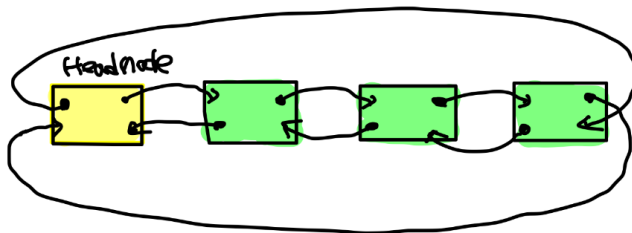
1829008 김민영

## HW3-1 : 이중 연결리스트의 삽입

메인함수에서 더미노드를 만들어주었기 때문에 같은방식으로 구현할 수 있다.

1829008  
김민영  
HW3-1

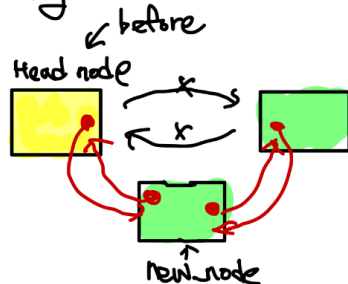
Doubly Linked List



yes!

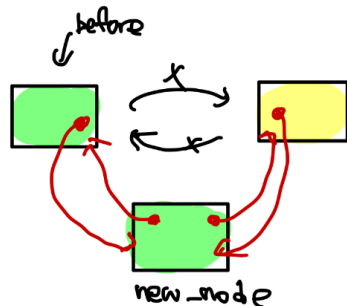
Head node (dummy node) 가 있어서 <IP의 "insert\_node"  
함수를 써서 beginning, end 에도 추가할 수 있다.

\* beginning



=> before 자리에 head node를 넣어줘야하면 된다.

\* end



=> before 자리에 그냥 마지막노드를 넣어주고  
그전에 end에 insert하게 됨

## HW 3-2 : Linked List merge (sort)

### 1. basic ideas and how you applied it into your code, analysis of your code(complexity)

dummy Node 를 사용해서 풀었다. Tail 은 항상 마지막 노드를 가리키므로 새 노드를 쉽게 추가할 수 있다.

Display 함수: 결과를 출력해준다.


Add\_first 함수: element 값을 넣어준다.

MoveListNode 함수: 노드를 이동시킨다.

SortedMerge 함수: 문제에서 구현하라고 하는 함수

time complexity : A 와 B 가 이미 정렬되어있는 상태에서  $n = n1(a) + n2(b)$  라고 한다면  $O(n)$ 이다.

### 2. result image

$a = \{1,2,5,10,15,20,25\}$   
 $b = \{3,7,8,15,18,30\}$    $c = \{1,2,3,5,7,8,10,15,15,18,20,25,30\}$

1



C:\WINDOWS\system32\cmd.exe

```
a : 1 -> 2 -> 5 -> 10 -> 15 -> 20 -> 25 -> NULL
b : 3 -> 7 -> 8 -> 15 -> 18 -> 30 -> NULL
c : 1 -> 2 -> 3 -> 5 -> 7 -> 8 -> 10 -> 15 -> 15 -> 18 -> 20 -> 25 -> 30 -> NULL
계속하려면 아무 키나 누르십시오 . . .
```

### HW3-3 : Simply Linked List 다양한 함수 구현

#### 1. basic ideas and how you applied it into your code, analysis of your code(complexity)

Lec04 ADT 페이지를 참고하여, head 와 tail 이 있는 상황으로 코드를 구현하였다.

Init : 초기화

createNode : 노드생성

add\_first : 첫번째에 추가

add\_last : 마지막에 추가

delete\_first : 첫번째에 추가

delete\_last : 마지막에 추가

get\_node\_at : 노드 가져옴

insert\_node : 노드삽입

add : 인덱스에 추가

is\_empty : 비었는지 확인

get\_length : 길이

remove\_node : 노드 제거

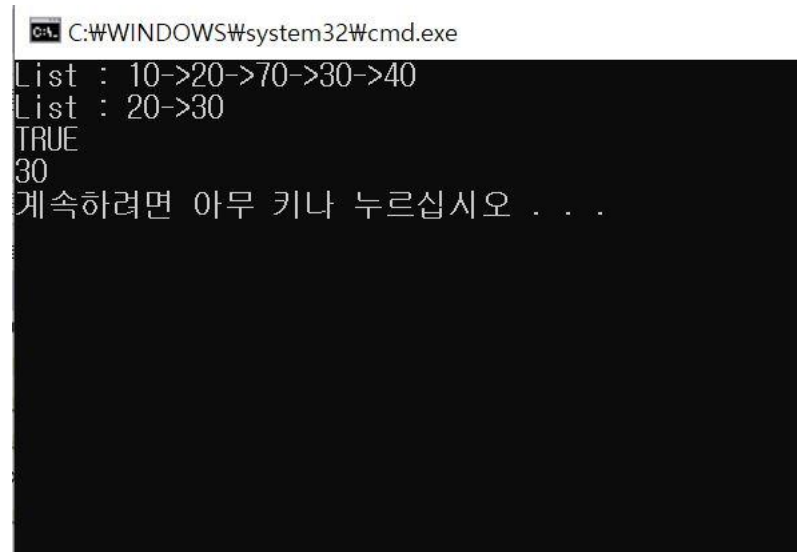
delete : 삭제

get\_entry : pos 위치 얻어오기

display : 출력해줌

is\_in\_list : 안에 들었는지 확인

#### 2. result image



```
C:\WINDOWS\system32\cmd.exe
List : 10->20->70->30->40
List : 20->30
TRUE
30
계속하려면 아무 키나 누르십시오 . . .
```