



Pervasive Elastic MetaLearning Laboratory
Department of Computer Engineering
Hongik University

HW4 실습

Linked List

연결 리스트 Linked List

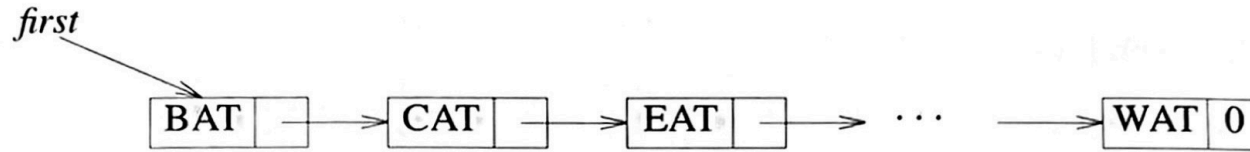


그림 4.2 연결 리스트를 그리는 일반적인 방법

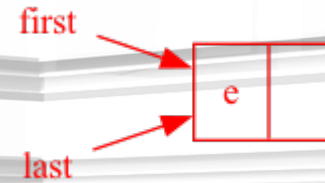
```
struct Node
{
    Node(int d = 0, Node* l = NULL) : data(d), link(l) { }
    int data;
    Node * link;
};
```

```
struct IntList
{
    IntList() { last = first = NULL; }
```

연결 리스트 Linked List : push_back()

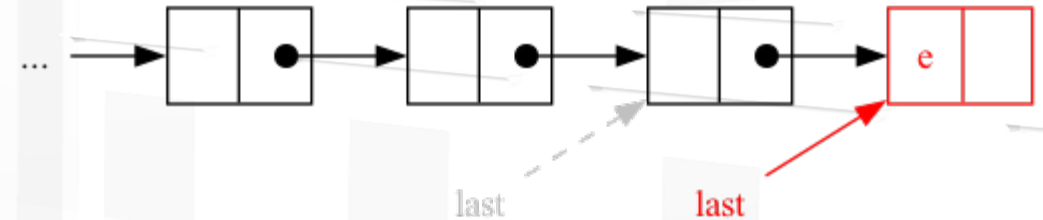
■ Empty List

- New node를 first, last로 align



■ Non-empty List

- Last가 가리키고 있는 노드 → New node
- New node를 last로 align

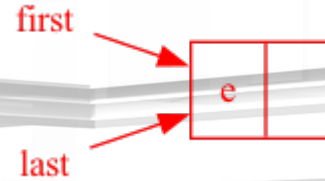


연결 리스트 Linked List : insert()

이때, insert()는 오름차순을 유지하며 올바른 위치에 노드를 삽입한다

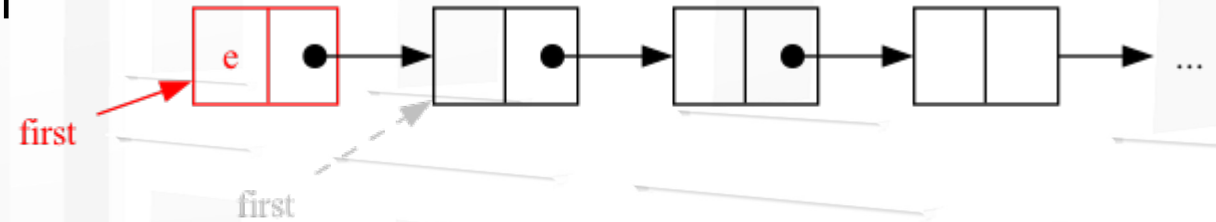
■ Empty List

- New node를 first, last로 align



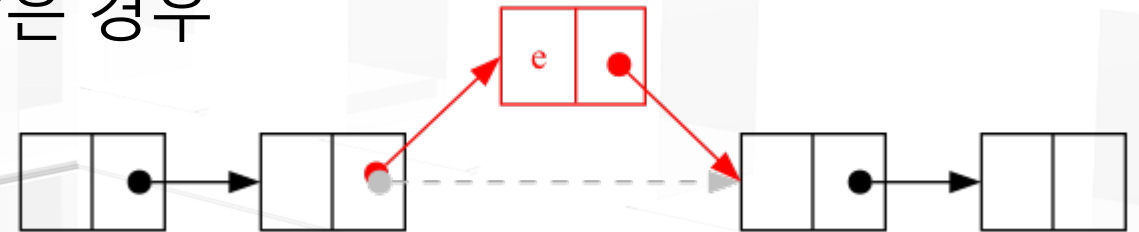
■ Insert Data가 First Data 보다 작은 경우

- New node → first가 가리키고 있던 노드
- New node를 first로 align



■ Insert Data가 First Data 보다 크거나 같은 경우

- Node 포인터로 리스트 순회 가능
- 추가할 위치 찾아서 New node 삽입



연결 리스트 Linked List : delete()

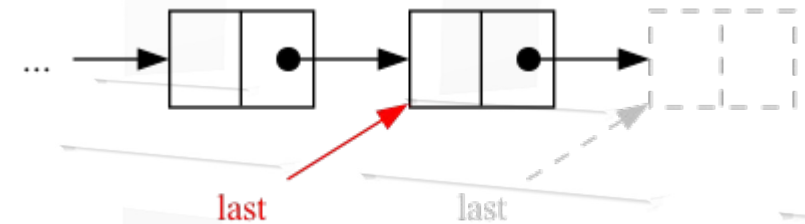
■ Deleting the Middle node

- Target 노드를 가리키고 있던 노드 → Target 노드가 가리키고 있던 노드



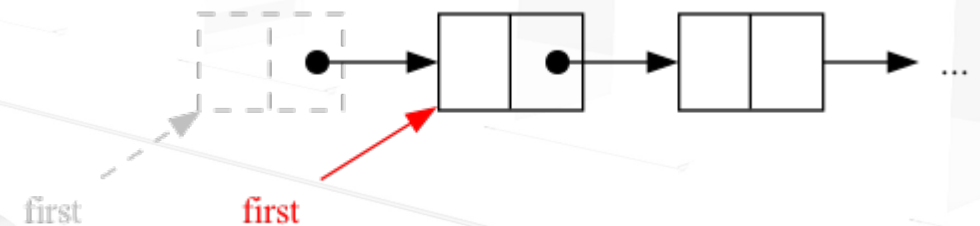
■ Deleting the Last node

- Target 노드를 가리키고 있던 노드 last로 align



■ Deleting the First node

- Target 노드가 가리키고 있던 노드 first로 align



연결 리스트 Linked List : insert()

Insert(), Delete()

예제 4.4: *first*와 *x*를 예제 4.3에서와 같다고 하자. *y*를 *x* 앞에 있는 노드(만일 있다면)를 지시하는 포인터라 하자. 그리고 만일 *y* == 0이기만 하면, *x* == *first*라고 하자. 함수 *Chain::Delete*(프로그램 4.5)는 체인으로부터 노드 *x*를 삭제한다. □

```
void Chain::Insert50(ChainNode *x)
```

```
{  
    if (first)  
        // x 다음에 삽입  
        x → link = new ChainNode(50, x → link);  
    else  
        // 공백 리스트에 삽입  
        first = new ChainNode(50);  
}
```

프로그램 4.4: 노드의 삽입

```
void Chain::Delete(ChainNode *x, ChainNode *y)
```

```
{  
    if (x == first) first = first → link;  
    else y → link = x → link;  
    delete x;  
}
```

실습 설명

- 기술서에 적혀있는 대로 list.cpp의 비어있는 부분을 작성할 것

```
struct IntList
{
    IntList() { last = first = NULL; }

    void Push_Back(int); // 리스트 뒤에 삽입 중복 허용
    void Push_Front(int); // 리스트 앞에 삽입 중복 허용
    void Insert(int); // 정렬되어 있다는 가정 하에 제 위치에 삽입
    void Delete(int); // 리스트의 원소 삭제
    Node *first; // 첫 노드를 가리킴
    Node *last; // 마지막 노드를 가리킴
};
```

질문

- pemds81718@gmail.com
- 간단한 구글링으로 알 수 있는 내용은 답변하지 않습니다.