

자료구조 Data Structure | 조행래

연결 리스트

이중 연결 리스트

학습 목표

- 이중 연결 리스트의 개념을 이해한다.
- 이중 연결 리스트에서 노드 추가 및 삭제 알고리즘을 이해한다.

1. 이중 연결 리스트의 개념

- 이중 연결 리스트(Doubly linked list)란?
 - 한 노드에 두 개의 link가 저장

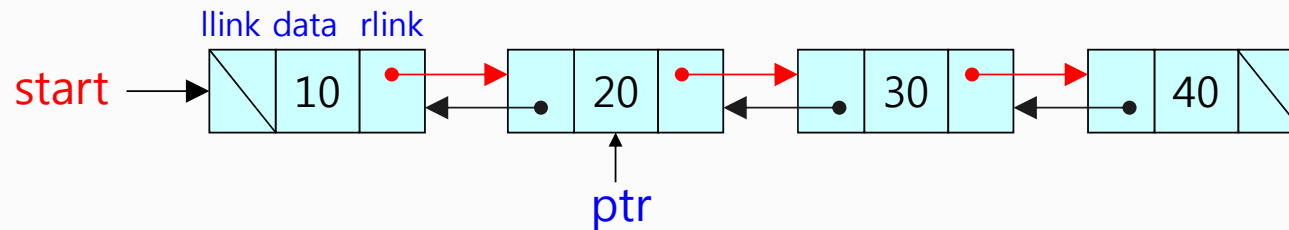
```
struct node {  
    struct node *llink;    // 이전 노드를 포인트  
    int      data;  
    struct node *rlink;    // 다음 노드를 포인트  
};
```

- 이중 연결 리스트는 양 방향으로 이동 가능
 - 단일 연결 리스트의 경우, 한 방향으로만 이동 가능

2. 이중 연결 리스트의 종류

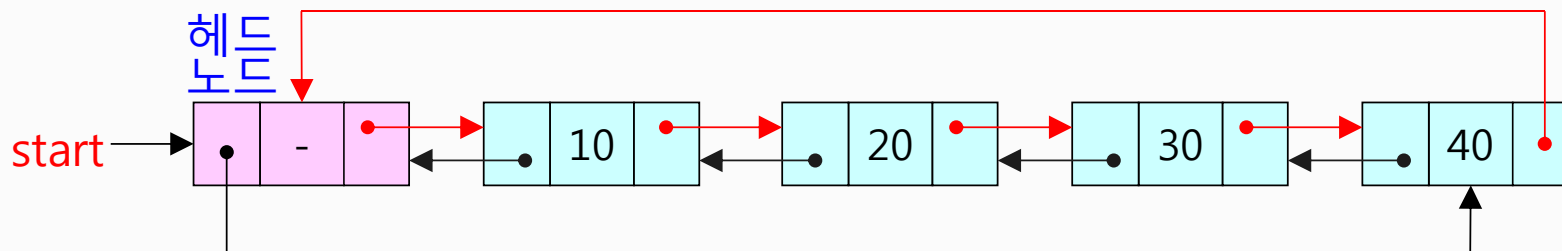
■ 체인

- 처음 노드의 llink와 마지막 노드의 rlink는 NULL

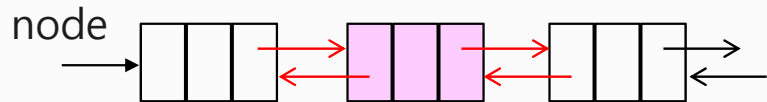
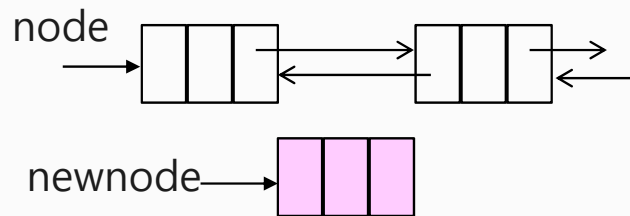


- $ptr = ptr \rightarrow llink \rightarrow rlink = ptr \rightarrow rlink \rightarrow llink$

■ 원형 이중 연결 리스트



3. 원형 이중 연결 리스트에 노드 추가

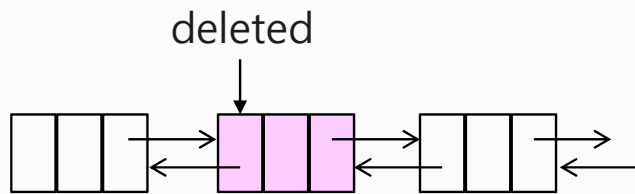


```
void dinsert(struct node *node,
struct node *newnode)
{
// newnode를 node의 오른쪽에
  추가
  newnode→llink = node;
  newnode→rlink = node→rlink;
  node→rlink→llink = newnode;
  node→rlink = newnode;
}
```

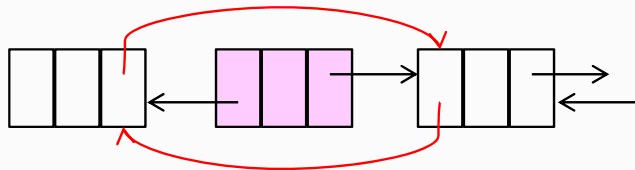
다른 문제들:

- (1) newnode를 node의 왼쪽에 추가
- (2) 원형 리스트가 아닌 이중 연결 chain
의 왼쪽과 오른쪽에 노드 추가

4. 원형 이중 연결 리스트에서 노드 삭제



```
void ddelete(struct node *deleted)
{
    deleted->llink->rlink = deleted->rlink;
    deleted->rlink->llink = deleted->llink;
    free(deleted);
}
```



다른 문제들:

- (1) deleted 노드의 이웃 노드 삭제
- (2) 원형 리스트가 아닌 이중 연결 chain에서 노드 삭제



요약 정리

- 이중 연결 리스트의 개념을 설명
- 이중 연결 리스트에서 노드 추가 및 삭제 알고리즘