

자료구조 Data Structure | 조행래

연결 리스트

연결 리스트를 이용한 스택과 큐

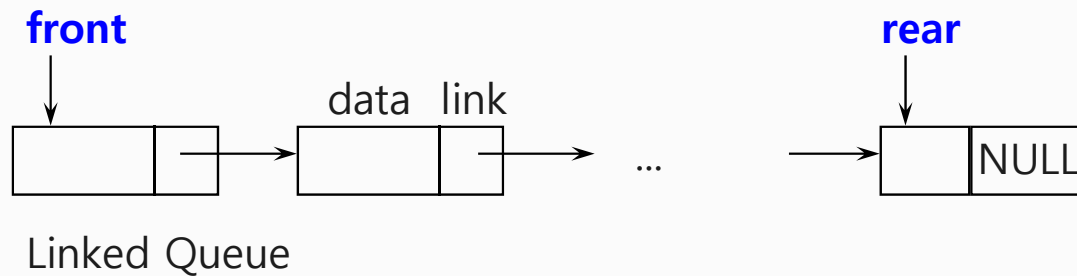
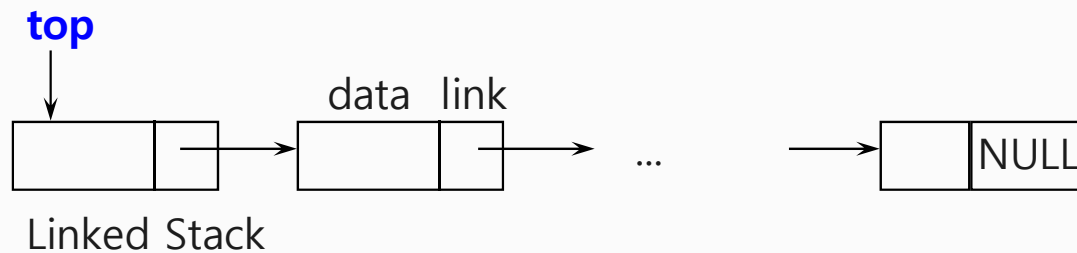
학습 목표

- 연결 리스트를 이용하여 스택을 구현할 수 있다.
- 연결 리스트를 이용하여 큐를 구현할 수 있다.
- 배열을 이용한 구현 방법과의 차이점을 이해할 수 있다.

1. 기본 개념

- 배열을 이용한 스택과 큐의 구현 방법의 문제점
 - 메모리 낭비
 - Stack Full의 발생 가능성

- 리스트를 이용한 스택과 큐의 구현



2. 연결 리스트를 이용한 스택의 구현

```
struct stack {  
    int data;  
    struct stack *link;  
};
```

```
struct stack *top;
```

(1) 초기 설정:

`top = NULL;`

(2) StackEmpty:

`top == NULL`

(3) StackFull:

메모리가 full이어서
malloc()의 결과로 NULL이
return될 때

스택에 추가

```
void push(int item)
{
    // 스택 top에 새로운 item 추가
    struct stack *tmp =
        (struct stack *) malloc(sizeof(struct stack));
    if (tmp == NULL) {    // memory full
        fprintf(stderr, "The memory is full\n");
        exit(1);
    }
    tmp->data = item;
    tmp->link = top;
    top = tmp;
}
```

1, 2, 3의 순서로 데이터가
추가될 때 stack의 모습?

스택에서 삭제

```
int pop()
{
    // 스택의 top이 가리키는 data를 삭제하여 return
    struct stack *tmp = top;
    int item;
    if (tmp == NULL) {    // top == NULL
        fprintf(stderr, "The stack is empty\n");
        exit(1);
    }
    item = tmp->data;
    top = tmp->link;
    free(tmp);
    return item;
}
```

3. 연결 리스트를 이용한 큐의 선언

```
struct Queue {  
    int data;  
    struct Queue *link;  
};
```

```
struct Queue *front, *rear;
```

(1) 초기 설정:

front = NULL;

(2) QueueEmpty:

front = NULL

(3) QueueFull:

메모리가 full이어서
malloc()의 결과로 NULL이
return될 때

큐에 추가

```
void addq(int item)
{
    // 큐의 rear에 새로운 element 추가
    struct Queue *tmp =
        (struct Queue *) malloc(sizeof(struct Queue));
    if (tmp == NULL) { // memory full
        fprintf(stderr, "The memory is full\n");
        exit(1);
    }
    tmp->data = item;
    tmp->link = NULL;
    if (front != NULL) rear->link = tmp;
    else front = tmp;
    rear = tmp;
}
```


큐에서 삭제

```
int deleteq()
{
    // 큐의 front가 가리키는 데이터를 삭제하여 return
    struct Queue *tmp = front;
    int item;
    if (tmp == NULL) {    // front == NULL
        fprintf(stderr, "The queue is empty\n");
        exit(1);
    }
    item = tmp->data;
    front = tmp->link;
    free(tmp);
    return item;
}
```



요약 정리

- 연결 리스트를 이용하여 스택을 구현하는 방법
- 연결 리스트를 이용하여 큐를 구현하는 방법