

자료구조 과제2 보고서

20191016 최정윤

```
int add(char pos_string[], int data)
```

원하는 곳에 새로운 트리노드를 생성하기 위하여 먼저 부모노드를 가르키는 포인터 p와 자식노드를 가르키는 포인터 c를 선언한다.

입력받은 문자열의 길이를 알기위해 strlen을 사용하여 pos_string의 문자열 길이를 n변수에 저장한다. p와 c는 모두 맨 처음에는 root노드를 가르키고 있다.

for반복문을 사용하여 pos_string 배열 인덱스를 하나씩 증가시키면서 각 인덱스의 문자 L이면 왼쪽 노드를 문자 R이면 오른쪽 노드를 가르켜서 포인터를 맨마지막 문자 바로 전까지 이동시킨다.

맨 마지막 문자가 입력될 때는 그 자리에 새로운 노드를 추가시켜주고 추가에 성공하면 1이 리턴되게 한다.

추가할 때 부모 노드가 비어 있다면 포인터를 이어줄 방법이 없으므로 추가가 불가능하다. 이때는 -1을 리턴시켜준다.

추가할 때 추가해야할 자리에 이미 데이터가 채워져 있다면 노드를 추가할 수 없다. 이때 -1을 리턴시켜준다.

```
int delete(char pos_string[])
```

원하는 노드를 삭제하기 위하여 먼저 부모노드를 가르키는 포인터 p와 자식노드를 가르키는 포인터 c를 선언한다.

입력받은 문자열의 길이를 알기위해 strlen을 사용하여 pos_string의 문자열 길이를 n변수에 저장한다. p와 c는 모두 맨 처음에는 root노드를 가르키고 있다.

for반복문을 사용하여 pos_string 배열 인덱스를 하나씩 증가시키면서 각 인덱스의 문자 L이면 왼쪽 노드를 문자 R이면 오른쪽 노드를 가르켜서 포인터를 맨마지막 문자 바로 전까지 이동시킨다.

맨 마지막 문자가 입력될 때는 그 자리의 노드에 NULL을 저장해주고 free시켜준다. 삭제에 성공하면 1이 리턴되게 한다.

삭제할 노드의 자식노드가 있다면 -1을 리턴시켜준다.

삭제할 노드의 부모노드가 없다면 -1을 리턴시켜준다.

```
void preorder(node_t* node)
```

전위순회는 (루트노드) -> (왼쪽노드) -> (오른쪽노드) 순으로 순회하며 값을 출력한다.

재귀함수를 이용하여 이 순서대로 data값이 출력되게 한다.

만약 루트노드가 비어있다면 그냥 리턴시켜준다.

```
void inorder(node_t* node)
```

중위순회는 (왼쪽노드) -> (루트노드) -> (오른쪽노드) 순으로 순회하며 값을 출력한다.

재귀함수를 이용하여 이 순서대로 data값이 출력되게 한다.

만약 루트노드가 비어있다면 그냥 리턴시켜준다.

```
void postorder(node_t* node)
```

후위순회는 (왼쪽노드) -> (오른쪽노드) -> (루트노드) 순으로 순회하며 값을 출력한다.

재귀함수를 이용하여 이 순서대로 data값이 출력되게 한다.
만약 루트노드가 비어있다면 그냥 리턴시켜준다.