

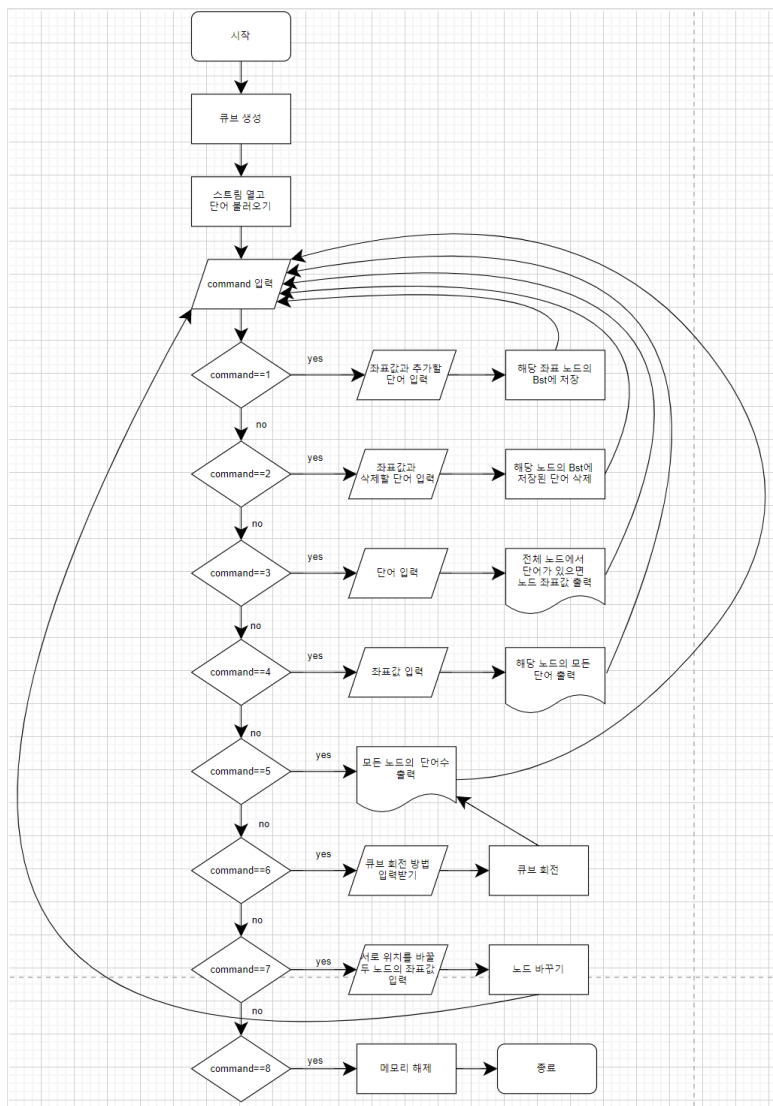
OOP Project 보고서

2019202050 이강현

1. Introduction

이번 프로젝트는 linked list를 활용하여 3D linked list 형식인 큐브구조를 노드를 통해 구현하고 각 노드에 BST구조를 활용한 단어 저장 그리고 노드에 좌표를 달아 좌표값을 통한 접근 수정이 가능하도록 구현한다. 또한 실제 큐브가 움직이는 것처럼 회전도 가능하게끔 구현한다. Linked list의 응용이며 동적할당, 포인터의 적절한 사용, 조건문의 분기등 많은 개념들이 복합적으로 연결되어 있어 개념의 정확한 이해를 확인할 수 있는 프로젝트이다.

2. Flowchart



각각의 명령에 따라 큐브는 위의 플로우차트와 같이 행동한다.

명령은 1부터 8까지 insert,delete,find,print,print_all,turn,exchange,exit의 행동을 취한다.

3. Algorithm

-큐브 구성방법-

초반 큐브를 구성하는 과정은 생성자를 통해 1D linkedlist로 27개의 노드를 이어 붙인 후 search1D함수로 좌표를 단서로 노드를 찾고 createcube함수를 이용해 해당 좌표가 0이면 plus만 연결,1이면 minus,plus연결, 2이면 minus만 연결하도록 하고 for문 중첩문으로 반복하여 3차원 큐브 구조로 구현하였다.(큐브형태로 모든 명령을 수행하는 것이 목적이므로 이 과정은 초반 큐브구조 구성시에만 사용한다.)

-각 좌표 접근법-

큐브를 완성한 후 search3D함수를 통해 좌표값을 인자로 받으면 해당 노드 포인터를 반환해주도록 하여 각 노드에 쉽게 접근할 수 있도록 하였다.

-BST에서 단어 수정방법-

Insert 구현은 BST의 노드의 헤드라는 BST전용 포인터를 통해 헤드부터 출발하며 아래로 내려가면서 strcmp내장함수를 통해 단어들간 대소를 비교하며 저장하고 delete 구현은 strcpy를 추가로 사용하여 같은 단어를 만났을 때 삭제하도록 구현하였다.

-find 구현방법-

Find명령은 locationprint 반환형이 bool형인 함수를 통해 단어를 찾고 findword라는 함수에서 위 함수를 사용하여 locationprint가 1을 반환하면 해당 좌표를 출력하도록 구현하였다.

-print 구현방법-

좌표값을 인자로 받고 BST에서 전위 순회,후위 순회,중위 순회를 모두 각각 함수

로 만든 후 사용하였다.

-print_all 구현 방법-

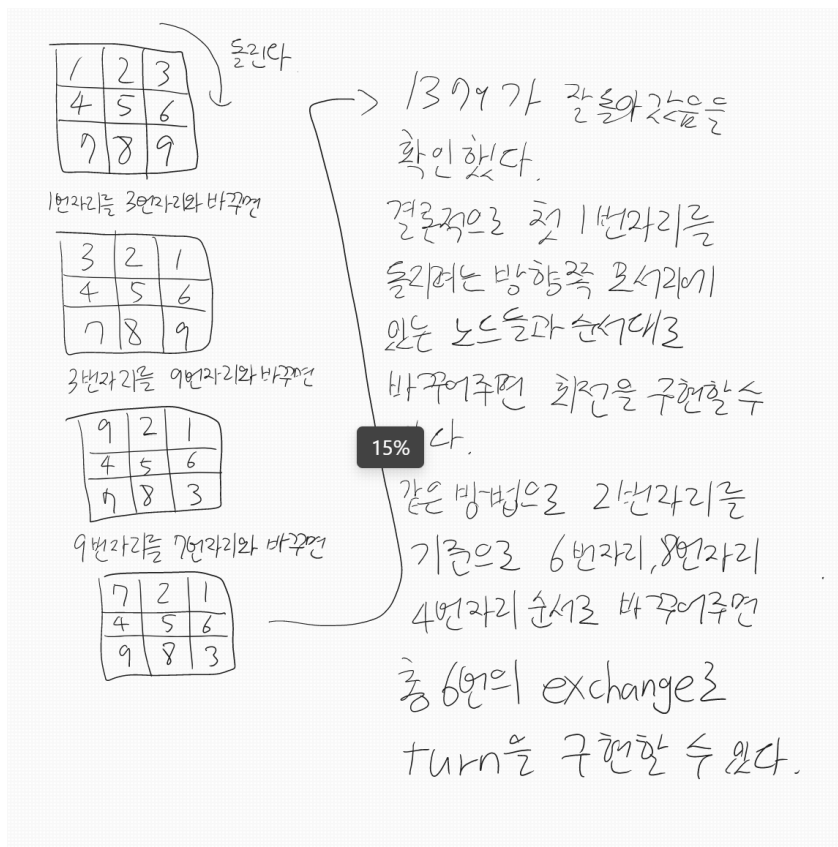
해당 명령은 모든 노드에 저장되어있는 단어 개수를 출력해야하기 때문에 wordcount라는 함수를 만들고 BST에서 순회를 통해 NULL이 아닌 단어에 접근할때마다 num값을 증가시켜 함수가 마무리되면 num을 출력하도록 구현했다.

-exchange 구현 방법-

노드간 해당 노드의 좌표를 바꿔주고 각 노드를 가리키는 노드들의 포인터또한 바꿔주며 해당 노드가 주변노드를 가리키는 포인터 또한 바꿔주어 exchange를 구현하였다. 포인터를 바꿔주는 과정에서 서로 인접한 노드일 경우 예외사항을 확인할 수 있었다. 따라서 temp노드를 하나 일시적으로 생성하여 노드를 바꾼 후 생기는 예외사항을 처리했다.

-turn 구현 방법-

Turn은 전면,측면,뒷면으로 조건을 나누고 시계방향과 반시계방향일때를 조건으로 나누고 exchange를 6번사용하여 구현했다.



-memory해제 구현-

Search3D함수를 통해 BST를 해제하고 각 노드를 해제했는데

Search3D함수는 cube클래스의 멤버변수인 center(1,1,1)에서 시작하여 x,y,z좌표를 증감하여 각 노드에 접근하기에 메모리를 해제할 때 search3D함수를 사용하려면 center에서 먼 쪽부터 해제를 진행해야한다. 따라서 (0,0,0)이나 (2,2,2)처럼 먼쪽 부터 차례대로 해제하도록 구현하였다.

Result & Verification

```
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 2 0 0
preorder : process bridge battle straw show
inorder : battle bridge process show straw
postorder : battle bridge show straw process
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 1 2 0 0 princess
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 2 0 0
preorder : process bridge battle princess straw show
inorder : battle bridge princess process show straw
postorder : battle princess bridge show straw process
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 3 princess
200
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 1 0 1 0 princess
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 3 princess
010
200
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 2 2 0 0 princess
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 2 0 0
preorder : process bridge battle straw show
inorder : battle bridge process show straw
postorder : battle bridge show straw process
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 3 princess
010
```

4번 명령어로 (2,0,0)에 해당하는 노드의 단어를 확인하고 1번 명령어로 단어를 추가했을 때 들어간 것을 확인할 수 있었다.

3번 명령어로 단어를 찾았을 때 (2,0,0)에만 존재했으나 (0,1,0)에도 추가한 이후 3번명령어에서 (0,1,0)에도 princess가 있음을 확인했다.

2번명령어로 (2,0,0)의 princess를 지우니 4번명령어로 단어를 확인했을 때 사라진 것을 확인했고 3번명령어도에 (2,0,0)이 사라짐을 확인했다.

```

6 4 4
2 2 5
5 4 4

4 4 4
2 1 1
3 2 6

5 2 4
2 4 3
3 10 5

Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 6 0 0 0
5 2 6
4 2 4
4 5 4

4 4 4
2 1 1
3 2 6

5 2 4
2 4 3
3 10 5

Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 6 1 0 0
6 4 4
4 2 4
4 5 4

2 4 2
2 1 1
3 2 6

5 4 5
2 4 3
3 10 5

Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 6 2 0 0
4 4 4
3 2 4
3 5 4

4 4 2
2 1 1
2 2 6

6 4 5
2 4 3
5 10 5

```

Turn 구현결과

전면(0) 시계방향(0) $y=0(0)$ 인 면 회전

윗면(1) 시계방향(0) $z=0(0)$ 인 면 회전

측면(1) 시계방향(0) $x=0(0)$ 인 면 회전

```
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 7 0 0 0 2 2 2
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 5
5 4 4
3 2 4
3 5 4

4 4 2
2 1 1
2 2 6

6 4 5
2 4 3
5 10 4

Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 7 0 0 0 0 0 1
Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 5
4 5 4
3 2 4
3 5 4

4 4 2
2 1 1
2 2 6

6 4 5
2 4 3
5 10 4

Enter Any Command(1 : insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 8
C:\Users\Wd1rkd\바탕 화면\과제\이강현\과제\설W코드W2-1_project\Wx64\Debug\W2-1_project.exe(프로세스 17568개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요....
```

Exchange 구현 결과

4. Conclusion

위 프로젝트를 구현하면서 다양한 명령에 대한 코드의 흐름과 클래스를 여러개 생성도 해보고 여태껏 배운 것에 대한 총 집합체를 만들어 낸 것 같아 뿌듯하였다. 포인터의 사용에도 능숙해졌고 Linked list의 사용도 자주 사용하니 자연스럽게 다룰 수 있게 되었다. 코드를 구현하는 과정에서는 exchange가 가장 어려웠는데 서로 인접한 노드를 바꾸는 과정이 어려웠다. 예시로 (0,0,0)을 가리키는 노드 포인터 A와 (0,0,1)을 가리키는 노드포인터 B가 있을 때 둘끼리 서로 좌표값을 바꾸더라도 A->xplus가 B이고 B->xminus가 A이기 때문에 서로 바뀌는 현상이 생겼고 주변포인터들은 상대를 가리키게하여 서로 포인터도 바꾸는 과정에서 오류가 발생하여 결국 인접한 노드일 때는 다 바꾼 후 포인터에 제대로 된 값을 넣어주는 것이 필요했다. 처음에는 왜 바뀌는지조차 이해하지 못했지만 코드를 구현하려고 노력하는 과정속에서 포인터에 충분히 익숙해졌다고 생각했지만 아직 부족함을 느낄 수 있었다. 가장 인상깊었던 순간은 큐브 구조를 만들어 낸 createcube함수와 center에서 접근하여 x,y,z값을 한번씩만 바꿔주면 모든 노드에서 접근할 수 있게끔 만든 search3D 함수 turn을 구현할 때 exchange를 6번만 사용한 그러한 생각들을 직접 구현하였고 작동하는 것들이 흥미로웠다. 생각하는 것을 코드화하는 이런 실력들을 그동안의 과제로 키워냈고 프로젝트에서 확실하게 보여준 것 같아 프로젝트가 준 영향이 컸다고 생각한다.

