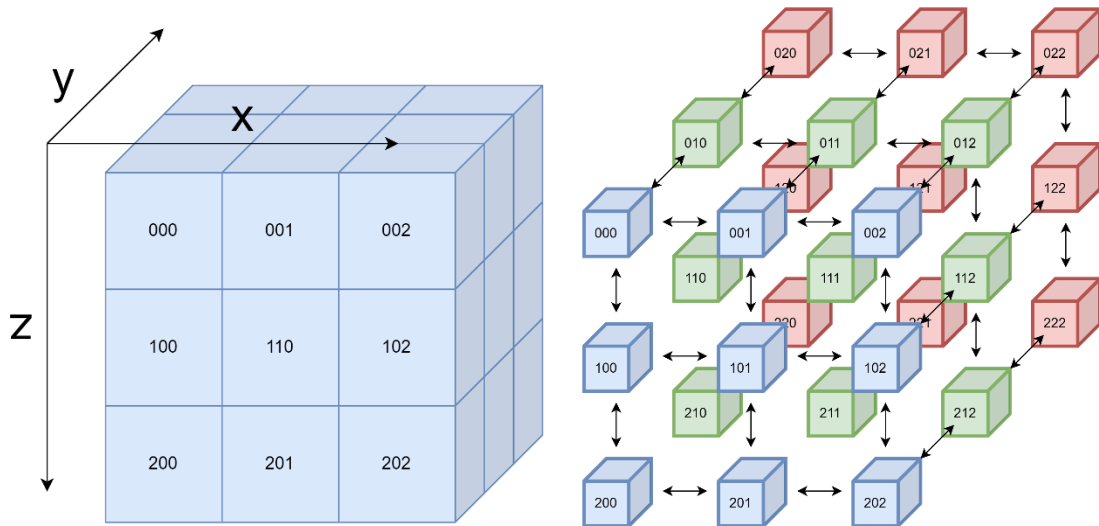


객체지향프로그래밍

Project

● 프로젝트 개요

프로젝트는 각 면이 회전할 수 있고 Block끼리 교환할 수 있는 3x3x3크기의 3D 큐브를 구성해 큐브 내부에 단어를 저장하는 프로그램의 구현을 목표로 한다. 큐브는 27개의 Block이 3D Linked List로 연결되어 Figure 1과 같이 구성되어 있으며 큐브의 각 면은 회전이 가능하다. 또한 큐브의 각 Block은 내부에 Binary Search Tree(BST)를 가지고 있어 BST를 이용해 단어의 저장, 삭제, 탐색, 출력이 가능하다.



<Figure 1>

● 데이터의 구조

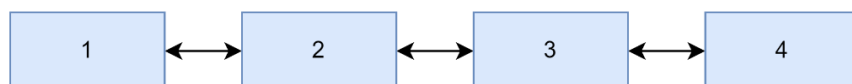
프로젝트는 3D Linked List를 이용해 27개의 Block을 연결하며 27개의 Block은 각각의 내부에 BST를 가지고 있다.

✓ 3D Linked List

3D Linked List는 각 Node가 6개의 방향(상,하,좌,우,앞,뒤)로 연결되어 있는 구조이다. 연결이란 각 Node가 가지고 있는 Pointer가 다른 Node를 연결하고 있는 것을 의미한다.

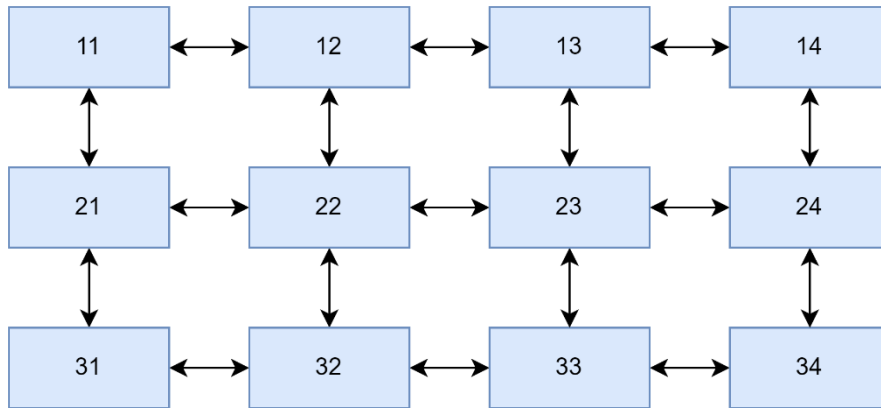
3D Linked List는 2D Linked List를 쌓아 구성할 수 있으며 2D Linked List는 1D Linked List들을 평행하게 펼친 후 연결하는 방법으로 구성할 수 있다.

Figure 2는 1D Linked List에 대한 그림으로 1D Linked List는 그림과 같이 각각의 Node(그림의 사각형)들이 서로 일렬로 연결되어 있는 구조를 말한다. 이 때 Node는 좌, 우의 Node들을 가리키는 포인터를 가지고 있어 각 Node를 가리킨다.



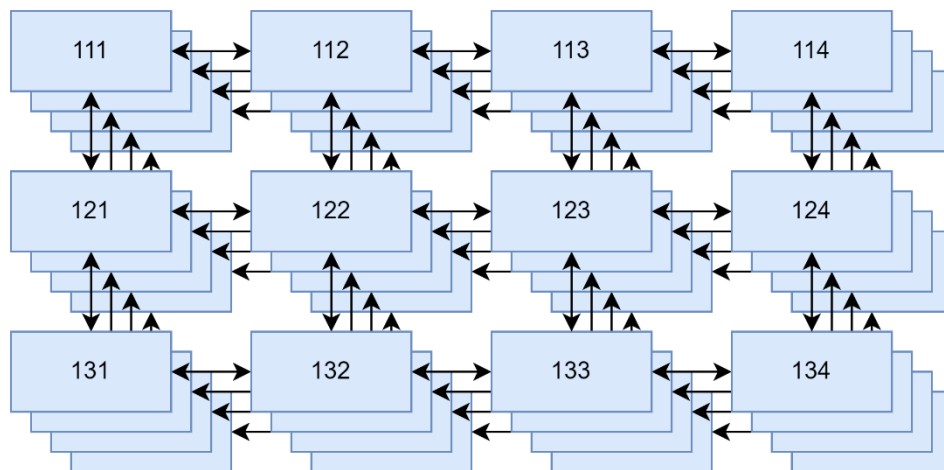
<Figure 2>

Figure3은 1D Linked List를 평행하게 펼친 후 서로 연결해 2D Linked List를 구성한 모습이다. 2D Linked List는 1D Linked List의 Node들이 서로 상,하로 연결되는 구조로 구성되며 2D Linked List의 각 Node들은 상,하,좌,우의 Node들을 가리키는 4개의 포인터를 가지고 있다.



<Figure 3>

Figure4는 2D Linked List를 쌓은 후 서로 연결해 3D Linked List를 구성한 모습이다. 3D Linked List는 2D Linked List의 Node들이 서로 앞, 뒤로 연결되는 구조로 구성되며 3D Linked List의 각 Node들은 상,하,좌,우,앞,뒤의 Node들을 가리키는 6개의 포인터를 가지고 있다.



<Figure 4>

✓ Binary Search Tree

Binary Search Tree는 한 개의 Node가 2개의 Child Node를 가지는 구조로 이루어진 자료 구조이다. Node의 Child Node는 각각 Left, Right로 분류하며 Left는 Node보다 작은 값의 Node를, Right는 Node보다 큰 값의 Node를 Child로 가진다.

BST는 Node를 삽입 시 최초의 Root Node부터 시작해 Node보다 작으면 Left, 크면 Right Child로 이동하며 다음 Node의 설치 장소를 찾으며 나아갈 방향에 Child Node가

없을 경우 그 자리에 삽입된다.

또한 BST는 Node 삭제 시 BST의 구조를 그대로 유지해야 하므로 아래 3가지 경우를 고려해서 Node를 삭제한다.

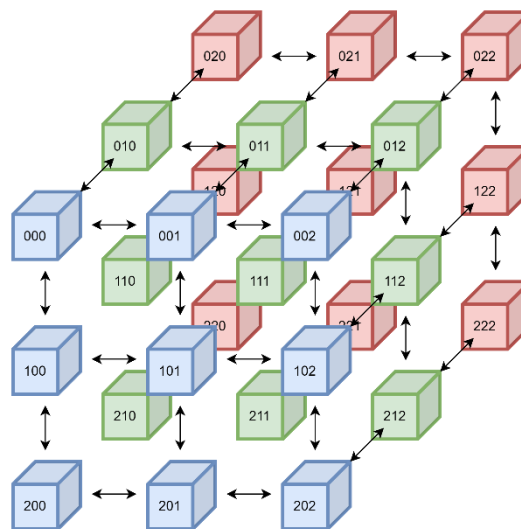
- Child가 없는 Node의 경우: Node만 삭제
- Child가 1개인 Node의 경우: Child Node를 삭제할 Node의 Parent와 연결 후 Node를 삭제
- Child가 2개인 Node의 경우: 삭제할 Node의 우측 Subtree 중 가장 왼쪽의 Node와 Node의 값을 바꾸고 Subtree의 가장 왼쪽 Node를 제거

● 프로젝트의 구성

프로젝트는 3개의 Problem으로 구성되어 있으며 3가지 Problem을 합쳐 하나의 프로젝트를 구성하며 EXIT 명령을 받기 전까지 Command를 입력 받아 동작한다. 세부 배점은 채점기준 참고

✓ Problem 1 (20점)

프로젝트 시작 시 3x3x3 크기의 3D Linked List로 구성되어있는 3D 큐브를 구성한다. 3D 큐브의 각 Block은 Figure 4와같이 상,하,좌,우,전,후를 서로 포인터를 이용해 연결되어 있으며 큐브의 각 Block은 전면 좌상단을 영점으로 (z,y,x)의 좌표값을 가지고 있다.(Block 들은 3d Linked List 외 다른 구조로 접근,삭제하지 않는다. Array에 block들을 넣고 저장해 사용하거나 하지 않는다)

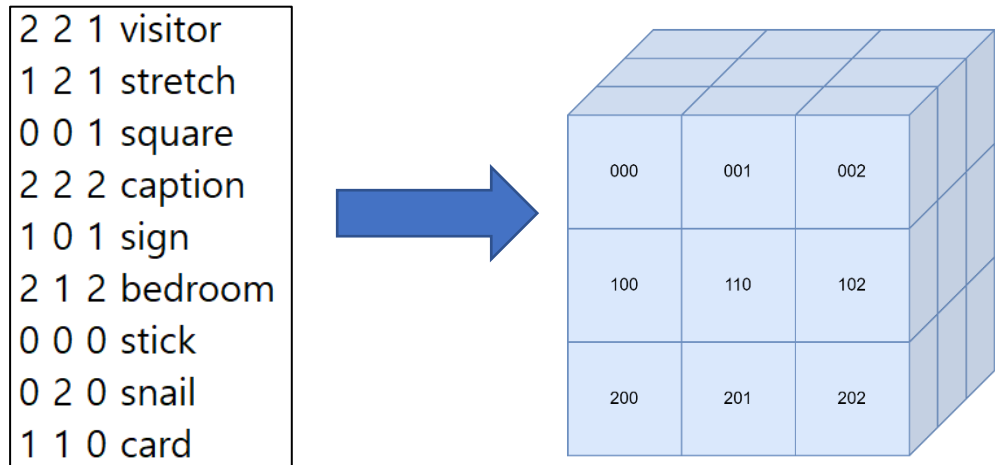


<Figure 5>

✓ Problem 2 (40점)

WordBook.txt파일을 읽어 3x3x3 3D 큐브의 각 Block에 저장한다. 각 Block은 하나의 BST 구조를 가지고 있으며 WordBook.txt은 줄마다 'z좌표 y좌표 x좌표 단어'로 구성되어 있어 (Figure 6 참고) WordBook.txt의 각 줄에 적혀져 있는 z좌표, y좌표, x좌표에 위치한

Block의 BST에 단어를 사전식 정렬 순으로 저장한다.



<Figure 6>

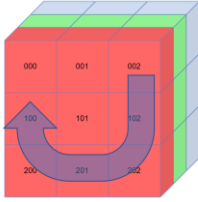
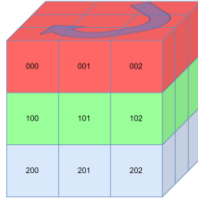
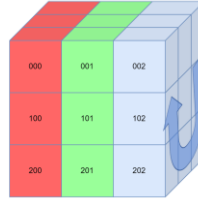
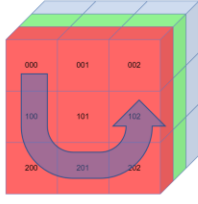
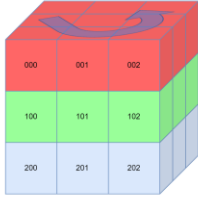
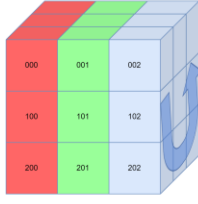
프로젝트는 3D Cube와 BST에 대해서 Table 1의 Insert, Delete, Find, Print, Print_All 명령어를 수행할 수 있어야 한다. (모든 명령어에서 단어의 대,소문자는 구별하지 않는다)

✓ **Problem 3 (40점)**

3D 큐브를 Turn 명령어에 따라 각 면을 회전시키고 Exchange 명령어에 따라서 Block의 위치를 교환한다. Turn은 Table 1에서 설명하고 있는 바와 같이 전면, 뒷면, 측면에 대해서 시계 방향, 반시계방향으로 면의 회전을 수행해야 하며(Figure 7), Exchange는 2개의 Block의 위치를 교환해야 한다. 또한 Turn과 Exchange의 수행 후에도 3D Cube의 모든 Block간 연결은 올바르게 연결되어 있어야 하며 반복해서 수행해도 오류가 없어야 한다.

✓ **Command Table**

명령어	사용법	사용 예시 및 기능설명
Insert	1 z y x 단어	<p>사용 예: 1 0 0 0 OOP</p> <p>(z,y,x) 좌표의 Block에서 단어를 삽입한다.</p> <p>단어는 Block의 BST에 단어를 사전식 정렬을 기준으로 삽입하며 만약 같은 단어가 있을 경우에는 삽입하지 않는다</p>
Delete	2 z y x 단어	<p>사용 예: 1 0 0 0 OOP</p> <p>(z,y,x) 좌표의 Block에서 단어를 삭제한다.</p> <p>Block의 BST에서 단어를 탐색하고 단어가 존재할 경우 단어를 삭제한다. 단어가 삭제될 경우에도 BST의 사전식 정렬은 유지되어야 하며 단어가 BST에 없을 경우 명령을 무시한다.</p>
Find	3 단어	<p>사용 예: 3 OOP</p> <p>단어를 가지고 있는 모든 큐브의 좌표를 출력한다.</p>

		3D큐브의 모든 Block의 BST에서 단어를 탐색해 단어가 있는 Block들의 좌표를 모두 출력한다. 좌표의 출력 순서는 임의로 한다.
Print	4 z y x	<p>사용 예: 4 0 1 0</p> <p>(z,y,x) 좌표의 Block에 저장된 모든 단어를 출력한다. 단어는 Pre-order, Post-order, In-order 3가지 방식 모두 사용하여 출력한다.</p>
Print_All	5	<p>사용 예: 5</p> <p>모든 Block에서 각 Block이 저장하고 있는 단어의 개수를 출력한다. 출력 순서는 아래와 같다.</p> <p>(z,y,x) 기준</p> <pre> 000 001 002 100 101 102 200 201 202 010 011 012 110 111 112 210 211 212 020 021 022 120 121 122 220 221 222 </pre>
Turn	6 A B C	<p>사용 예: 6 2 2 1</p> <p>큐브의 한 면을 회전한다. 큐브의 회전의 경우의 수는 전면, 측면, 후면과 각 면의 양방향 회전 6개의 경우가 있으며 3x3x3큐브 이므로 모든 경우의 수가 3개의 면에 대해서 회전이 가능해야 한다.</p> <div style="display: flex; flex-wrap: wrap; justify-content: space-around;"> <div style="text-align: center;">  <p>6 0 0 y(0:적 1:녹 2: 청)</p> </div> <div style="text-align: center;">  <p>6 1 0 z(0:적 1:녹 2: 청)</p> </div> <div style="text-align: center;">  <p>6 2 0 x(0:적 1:녹 2: 청)</p> </div> <div style="text-align: center;">  <p>6 0 1 y(0:적 1:녹 2: 청)</p> </div> <div style="text-align: center;">  <p>6 1 1 z(0:적 1:녹 2: 청)</p> </div> <div style="text-align: center;">  <p>6 2 1 x(0:적 1:녹 2: 청)</p> </div> </div> <p style="text-align: center;"><Figure 7></p>

		<p>명령어의 A는 전면(0) 윗면(1) 측면(2)을 나타내며 B는 시계방향(0) 반시계방향(1)을 나타내고 C는 A가 0인 경우 y, 1인 경우 z, 2인 경우 x를 나타낸다. 또한 회전 후 회전 결과를 출력한다.</p>
Exchange	7 z1 y1 x1 z2 y2 x2	<p>사용 예: 7 0 0 0 2 0 2</p> <p>큐브의 2개의 Block의 위치를 교환한다. 이 때 Block의 내부의 BST가 아닌 큐브의 Block을 교환해야 한다.</p>
Exit	8	프로그램 종료

<Table 1>

● 결과화면

✓ Insert 및 Print, Find 명령어

```

Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 0 0 0
Preorder : stick sheep dime orange vein thunder
Postorder : orange dime sheep thunder vein stick
Inorder : dime orange sheep stick thunder vein
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 1 0 0
Preorder : cent ghost
Postorder : ghost cent
Inorder : cent ghost
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 1 0 0 0 ghost
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 0 0 0
Preorder : stick sheep dime orange ghost vein thunder
Postorder : ghost orange dime sheep thunder vein stick
Inorder : dime ghost orange sheep stick thunder vein
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 3 ghost
000
100
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 1 0 1 0 ghost
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 3 ghost
000
100
010

```

1,2번 command: Print 명령어를 통해 (0,0,0)과 (1,0,0) Block 내의 Word를 출력

3번 command: Insert 명령어를 통해 (0,0,0) Block에 ghost 삽입

5번 command: Find 명령어를 통해 모든 Block 중 ghost가 존재하는 Block들을 탐색,
출력

6,7번 command: (0,1,0)에 ghost를 삽입하고 다시 Find 명령어로 ghost를 탐색

✓ Delete 명령어

```

Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 2 1 0 0 ghost
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 1 0 0
Preorder : cent
Postorder : cent
Inorder : cent

```

1번 command: Delete를 통해 (1,0,0) Block에서 ghost 제거

2번 command: 1번 command의 Delete를 통해 이전 그림에 존재하던 ghost가 사라짐

✓ **Print_All, Turn 명령어**

```

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

3      2      3
4      1      2
4      1      6

5      2      3
2      4      10
5      3      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
4      4      6
5      2      2
4      4      5

3      2      3
4      1      2
4      1      6

5      2      3
2      4      10
5      3      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      3      3
5      2      2
4      4      5

4      2      2
4      1      2
4      1      6

4      3      5
2      4      10
5      3      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      3      3
4      2      2
5      4      5

5      2      2
4      1      2
2      1      6

6      3      5
4      4      10
4      3      5

```

1번 command: 각 Block이 저장한 단어의 개수를 출력

2번 command: 전면(0) 시계방향(0) y=0(0)인 면 회전

3번 command: 윗면(1) 시계방향(0) z=0(0)인 면 회전

4번 command: 측면(1) 시계방향(0) x=0(0)인 면 회전

✓ **Exchange 명령어**

```

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

3      2      3
4      1      2
4      1      6

5      2      3
2      4      10
5      3      5

Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 0 0 0
Preorder : stick sheep dime orange vein thunder
Postorder : orange dime sheep thunder vein stick
Inorder : dime orange sheep stick thunder vein
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 2 0 2
Preorder : ladybug doctor desk scent
Postorder : desk doctor scent ladybug
Inorder : desk doctor ladybug scent
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 0 2
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 5
4      2      5
4      2      4
4      5      6

3      2      3
4      1      2
4      1      6

5      2      3
2      4      10
5      3      5

Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 0 0 0
Preorder : ladybug doctor desk scent
Postorder : desk doctor scent ladybug
Inorder : desk doctor ladybug scent
Enter Any Command(1 : Insert, 2 : Delete, 3 : Find, 4 : Print, 5 : Print_All, 6 : Turn, 7 : Exchange, 8 : Exit) : 4 2 0 2
Preorder : stick sheep dime orange vein thunder
Postorder : orange dime sheep thunder vein stick
Inorder : dime orange sheep stick thunder vein

```

4번 command: (0,0,0) Block과 (2,0,2) Block을 교환

1,5번 command: 1번과 5번을 비교해보면 (0,0,0)Block과 (2,0,2) Block을 교환된 것을 볼 수 있음

2,3번 / 6,7번 command: (0,0,0) Block과 (2,0,2) Block의 단어가 교환된 것을 볼 수 있음

● **채점기준**

No	항목	점수
1	3D 큐브 구조 구현	20
2	각 Block내 BST 구현	10
3	Insert/Delete 명령어 수행	10
4	Find 명령어 수행	10
5	Print 명령어 수행	5
6	Print_All 명령어 정상 수행	5
7	Turn 명령어 정상 수행	30
8	Exchange 명령어 정상 수행	10
총점		100

● 주의사항 및 참고사항

1. 채점자는 소스코드를 수정하지 않습니다. (소스코드 수정이 필요할 시 0점)
2. 제출기한 외 제출은 받지 않습니다.
3. 3D Linked List와 BST는 반드시 제안서에 제안한 방식으로 동작해야 합니다.
4. 제안서에 명시된 내용 이외의 내용은 자유롭게 구현할 수 있습니다.
5. 정상 구현되지 않은 명령어에 대해서는 명령을 무시하게 구현해주세요.
6. 과제에서 제공하는 텍스트 파일은 예시 파일로 제공된 파일 외의 내용에 대해서도 제대로 동작해야 합니다.