

ICE2004 자료구조론 과제5

제 목

자료구조 H/W #5

보고서 작성 서약서

1. 나는 타학생의 보고서를 베끼거나 여러 보고서의 내용을 짜집기하지 않겠습니다.

2. 나는 보고서의 주요 내용을 인터넷사이트 등을 통해 얻지 않겠습니다.

3. 나는 보고서의 내용을 조작하지 않겠습니다.

4. 나는 보고서 작성에 참고한 문헌의 출처를 밝히겠습니다.

5. 나는 나의 보고서를 제출 전에 타학생에게 보여주지 않겠습니다.

나는 보고서 작성시 윤리에 어긋난 행동을 하지 않고 정보통신공학인으로서 나의 명예를 지킬 것을 맹세합니다.

2021년 12월 17일

학부 정보통신공학과

학년 2

성명 김민겸

학번 12201863



1. **개요**

텍스트이(가) 표시된 사진

자동 생성된 설명

expandExternal()과 removeAboveExternal() 함수를 바탕으로, BST를 완성하는 것이 과제이다.

BST의 주요 기능을 다음과 같이 세 가지로 정의하였고, 트리 형태로 출력하고 트리 순회 방식에 따라 다르게 출력하는 세 함수를 만들었다.

* Search 기능 : 입력한 Key값을 기준으로, 해당되는 Node를 찾는 기능이다.
* Find(const Key &) : finder() 함수를 호출해, 노드를 찾았을 경우 노드를 반환하는 함수
* Finder(const Key &, NodePtr) : 트리를 순회하며, 현재 노드의 key값과 찾을 노드의 key값을 비교해 반환하는 함수
* Insert 기능 : 추가할 노드의 key와 value를 입력 받아 트리에 Node를 추가하는 기능
* Insert(const Key &, const Value &) : node의 Key와 Value를 추가하는 함수. 이때 외부 노드일 경우 expandExternal() 함수를 호출하여 외부 노드를 확장해준다.
* Remove 기능 : 선택한 key값에 해당하는 노드를 트리에서 삭제하는 기능
* Remove(const Key &) : key값을 기준으로 찾은 노드에 대한 remover()함수를 호출하고, key가 없는 노드에 해당한다면 Error message를 발생시킨다.
* Remover(const NodePtr &r) : 찾은 노드에 대해서 r 노드를 우측 아래 노드의 값으로 대입시키고, removeAboveExternal() 함수를 호출하여 노드를 삭제한다.
* 출력 기능 : 문제의 조건에 맞게 트리, 혹은 내부 key들을 출력한다.
* Print\_BT() : Binary Search Tree의 형태로 콘솔에 출력한다.
* Print\_preorder(), preorder() : preorder 순회 방식으로 출력한다.
* Print\_postorder(), postorder() : postorder 순회 방식으로 출력한다.
* Print\_inorder(), inorder() : inorder 순회 방식으로 출력한다.

1. 구현상 특징
2. **Entry Class**

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 1 Key와 Value의 속성을 담은 Entry Class

처음에는 Binary Tree의 한 노드 당 배정되는 key와 value를 어떻게 표현할 지 고민하다, 자료구조론 교과서를 참고해 (key, value)의 pair를 가지는 Entry class를 설계하게 되었다. 코드의 다형성을 위해, template 키워드를 이용해 key와 value에 대한 자료형을 각각 Key, Value로 지정하였다. 그리고 내부 멤버 함수로는 생성자와 소멸자 이외에 Get, Set함수를 정의하였다.

텍스트, 화면, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이렇게 BinarySearchTree에 대한 인스턴스를 main함수에서 생성할 때, 원하는 자료형을 대입해 Entry 클래스를 자료형으로 가지는 BinarySearchtree를 설계할 수 있었다.

1. **BinarySearchTree Class**
2. Node class

텍스트이(가) 표시된 사진

자동 생성된 설명

자료구조론 교과서를 참고하여 Node 클래스를 구현하였다. 이때 Node 클래스는 BinarySearchTree 내부에 정의해, 같은 Template typename을 공유하면서도, 내부 함수가 Node에 대한 속성을 사용할 수 있도록 한다. 이 때 typedef를 이용해 NodePtr이라는 키워드로 Node의 포인터 자료형을 사용할 수 있도록 하였다.

② search

텍스트이(가) 표시된 사진

자동 생성된 설명

Key값을 기준으로 노드를 찾아 반환하는 함수인 find는 내부에서 finder을 호출해, 찾고자 하는 노드의 key값과 매개변수로 들어온 key를 비교해 노드를 반환한다. Node가 외부 노드일 경우에는 더 이상 트리를 순회할 수 없으므로 node를 반환하고 함수 호출을 끝낸다.

내부 노드일 경우는, 노드의 Key 값이 찾고자 하는 노드의 Key 값과 일치할 때까지 key와 node->key()의 대소 관계를 비교해가며 작을 경우에는 왼쪽으로, 클 경우에는 오른쪽으로 이동한다.

③ insert

텍스트이(가) 표시된 사진

자동 생성된 설명

insert 함수는 매개변수에 추가될 key, value 쌍을 대입해준다. Finder 함수를 호출해 현재 root 노드를 출력해준다. 외부 노드에 도달할 때까지 while 문 안에서 p 노드를 오른쪽으로 이동해주고, 외부노드에 도착다면 expandExternal 함수를 호출해 외부노드를 확장한다.

setKey, setValue 함수에 인자로 들어온 key, value를 넣어줌으로써 p 노드에 대한 세팅을 해주고 노드의 개수를 의미하는 size변수를 1 증가시켰다.

④ Remove

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 2 예외 처리 클래스

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 3 remove, remover 함수

텍스트이(가) 표시된 사진

자동 생성된 설명내부 노드일 경우 해당 노드를 기준으로 큰 오른쪽 노드 중 가장 작은 노드를 탐색해 외부 노드가 될 때까지 이동한다. 그렇게 찾은 노드 p의 parent의 key, value값을 r노드, 즉 삭제하려는 노드에 대입한다.

그림 4 removeAboveExternal 함수

그리고 외부 노드를 삭제하는 함수를 호출하여 p에 해당하는 노드와 그 노드의 자식을 포함해 삭제한다.

⑤ print 함수

텍스트, 스크린샷, 명판이(가) 표시된 사진

자동 생성된 설명

Preorder, postorder, inorder 함수는 각각의 트리 순회의 순서에 맞게 재귀호출을 수행하였다.

Node 클래스 내부에 선언한 멤버 함수 print()를 호출해, (key, value) 쌍을 출력하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그림 5 트리 형태로 출력하는 함수

이 함수의 경우에는, 트리가 왼쪽부터 채워지고, 재귀호출에 의해서 prefix에 isLeft(이후에 출력해야 할 노드가 남았는지 여부)에 따라 공백이 더해지는 형태이다. 따라서 Root노드부터 시작해 왼쪽으로 증가하는 모양으로 트리가 그려지게 된다.

main함수에서 Node 객체 자체를 만들어 매개변수에 넣을 수 없기 때문에, 함수를 여러 번 오버로딩해 print\_BT()를 호출하였을 때 순차적으로 나머지 두 함수가 호출되도록 설계하였다.

Node가 leftChild, rightChild로 이동하면서 nullptr가 되었을 때 함수가 종료되도록 중단점을 설정하였기 때문에, External 노드에서 더 이상 expand된 노드를 만나지 못했을 때 호출이 종료된다.

1. **실행화면 캡쳐 이미지**

텍스트이(가) 표시된 사진

자동 생성된 설명

1. **고찰**

호출 결과에서, 외부 노드에서 expand된 노드 중 왼쪽, 오른쪽 자식은 ‘0’으로 나타난다.

설계한 print함수에서는 nullptr가 되었을 때 중단점을 설정하였기 때문에 가장 마지막 level의 노드가 아닌 그 아래 default로 생성된 노드들이 출력되는 문제가 있었다.



그래서 node의 leftChild, rightChild에 대해서 중단점을 설정해주면 문제가 해결될 것이라고 생각했으나, 출력이 동일하게 나오는 것을 확인하였다. 0으로 임의로 세팅된 최하위 노드에 대한 출력 방지 방법을 더 알아보아야겠다고 생각했다.

이번 과제에서는 특히 Key, Value에 대한 template의 처리 부분에서 가장 곤혹을 겪었다.

텍스트이(가) 표시된 사진

자동 생성된 설명

교과서에 실린 이 코드를 통해서 힌트를 얻을 수 있었는데, typedef와 typename이 같이 사용된 경우에 대해서 제대로 해석을 하지 못해 어려움이 있었다.

그러다 StackOverflow에서 위 문제에 대해 설명을 한 부분이 있어 참고하여 해석하였더니 이 SearchTree의 멤버 변수는 다음과 같이 이해할 수 있었다.

typename은 컴파일러에게 Key가 어떠한 멤버가 아닌 Type임을 알린다.  :: (namespace)가 유형의 범위이며, Key는 E의 범위 안에 있음을 의미한다.

즉 이 SearchTree에서는 K라는 키워드를 E라는 typename의 범주에 속하는 Key라는 타입을 대체한다고 이해하였다.

이렇게 어려운 구문에 대해서 충분히 의문점을 가져보고 해결해보는 경험을 할 수 있었고, BinaryTree에 대해 더 깊게 알게되는 계기가 되었다.