DS\_Assignment\_3

201716472 컴퓨터공학과

최유진

1 레드 블랙 트리(Red-Black Tree)

1.1 2-3트리와 동등한 레드 블랙 트리

코드 설계(클래스 계층도) :

RBtree\_test.cpp

RBtree.cpp

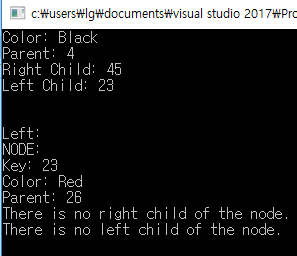
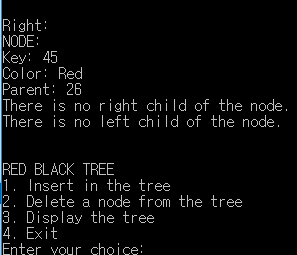
#include ”RBtree.cpp”

RBtree.cpp 에 node의 이름을 가진 구조체를 선언하였고, RBtree 클래스를 생성하였다. 클래스 안에는 RBtree(), Insert(), Insertfix(Nptr), Leftrotate(Nptr), Rightrotate(Nptr), Delete(), disp(), successor(Nptr), delfix(Nptr), Display(Nptr p) 함수를 선언하였다.

Insert 함수는 node를 선언하여 값들을 초기화 하였고 Insertfix() 함수를 사용하여 트리에 값을 넣는 역할을 하였다. Insertfix 함수는 Leftrotate와 Rightrotate 함수를 사용하여 RBtree의 색을 구분하여 설정하는 역할을 하였다. Delete함수는 while문을 사용하여 빼는 값에 따라 tree를 다시 조정하는 역할을 하였다. delfix 함수는 Leftrotate와 Rightrotate 함수를 사용하여 delete 함수를 사용할 때 색을 조정하는 역할을 하였다. Leftrotate 함수는 LChild 값과 RChild값과 Parent 값을 비교하여 교체하는 역할을 하였다. Rightrotate함수 또한 비슷한 역할을 하였다. successor 함수는 Nptr y를 선언하여 그 y 값에 RChild값과 LChild값을 받고 return 하는 역할을 한다. disp와 Display 함수는 tree를 출력하는 역할을 한다

RBtree\_test.cpp 에는 main() 함수를 만들고 그 안에 삭제 추가 보여주기 나가기를 추가하고 switch문을 사용하여 그에 따라 함수들을 실행하고 출력하였다.

실행 결과 :

1.2 2-3-4 트리와 동등한 레드 블랙 트리

코드 설계(클래스 계층도) :

#include ”RBtree\_234.cpp”

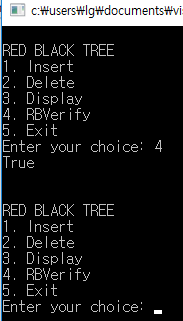
RBtree\_234.cpp

RBtree\_234\_test.cpp

RBtree\_234.cpp 1\_1번과 동일한 코드를 사용하였고 bool RBVerify 함수를 추가하였다. x가 1일 경우 true를 출력하고 다른 수일 경우 false를 출력하는 역할을 하였다.

RBtree\_234\_test.cpp의 경우 1\_1번과 동일하나 switch 함수에 RBVerify를 확인하는 경우를 추가하여 출력할 수 있도록 하였다.

실행 결과 :



2 트라이(Trie)

2.1.1 트라이를 이용한 Word count 계산 및 사용자 테스트

코드 설계(클래스 계층도) :

Trie.h

Trie.cpp

#include ”Trie.h”

Trie\_word\_count\_test.cpp

문제에 써있는 코드들을 모두 입력하였다.

3 네비게이터

3.1 네비게이터 구현

코드 설계(클래스 계층도) :

버튼 3개와 EditorPane을 사용하여 GUI를 설계하였다.

실행 결과 :

