7th assignment

2013011491

컴퓨터전공

안찬영

void rearrange(bTree b, bTree b2,int pos, int mid){

memmove(&b->keysAndKids.keys[pos+1],&b->keysAndKids.keys[pos],sizeof(\*(b->keysAndKids.keys)) \* (b->keysAndKids.numKeys-pos));

memmove(&b->keysAndKids.kids[pos+2], &b->keysAndKids.kids[pos+1], sizeof(\*(b->keysAndKids.keys)) \* (b->keysAndKids.numKeys-pos));

b->keysAndKids.keys[pos]=mid;

b->keysAndKids.kids[pos+1]=b2;

b->keysAndKids.numKeys++;

}

재배열 함수인데, 주석에 이미 설명이 모두 나와있는 것과 다름없다. 처음엔 하위 레벨에 있는 mid 값을 저장할 공간을 마련하기위해, keys배열에서 pos 기준으로 한 칸씩 메모리 이동한다. 그다음, b2를 연결할 공간을 만들기 위해, kids배열에서 pos+1기준으로 한 칸씩 메모리 이동한다,

그 다음 만들어진 공간에 mid를 대입해주고, b2를 연결한 뒤, 키 보유수를 하나 증가시킨다.

if(b->isLeaf) {

memmove(&b->keysAndKids.keys[pos+1], &b->keysAndKids.keys[pos], sizeof(\*(b->keysAndKids.keys)) \* (b->keysAndKids.numKeys-pos));

b->keysAndKids.keys[pos]=key;

b->keysAndKids.numKeys++;

삽입함수의 일부분인데, 마지막 잎 노드까지 왔을 때의 삽입 경우이다. 삽입할 위치 기준으로 key배열에서 한칸씩 메모리 이동해주고, key값을 저장한다. 그 다음 키 보유수를 증가시킨다.

<결과 1> <결과2>

