**자료구조-이진 탐색 트리 병합**

2020111340 우은지

**1. Environment**

visual studio 2019에서 작성한 소스코드이다.

**2. 추가된 함수**

int return\_getParentNode(BinSearchTreeNode\* pCurrentNode, int key)

- insert할 노드의 부모 노드를 찾아서 부모노드의 키를 출력해주는 함수

- key == pCurrentNode->key인 경우 중복된 키라고 출력해주진 않고 ret을 그냥 0으로만 함

(이 함수 다음으로 바로 insertDataBST()를 사용하기 때문에 insertDataBST() 함수 사용시 호출되는 getParentNode에서 중복된 키라고 출력해줌)

**3. main 함수 동작 과정**

1. 먼저 input 파일과 output 파일을 열어주고 파일이 제대로 열렸는지 검사함

2. nodes1, nodes2이라는 배열을 만들어서 엣지를 그대로 읽어서 저장

(ex 5 7이면 0번 인덱스에 5 저장, 1번 인덱스에 7 저장)

3. input 파일 첫 줄을 가장 먼저 읽어 엣지 수를 알아내서 첫 번째 트리의 엣지까지만 nodes1에 저장하기 위해서 input 파일 첫 줄을 읽어 first\_tree에 저장

(즉 first\_tree는 첫 번째 트리의 엣지의 수를 의미)

4. while문으로 파일을 한 줄씩 읽어오는데 읽어온 한줄을 공백 기준으로 나눠서 배열에 순서대로 저장

5. 두 번째 트리도 3, 4번의 과정 똑같이 해줌

6. t3가 될 트리를 생성해줌

7. 추가될 엣지, 삭제될 엣지의 개수 출력을 위해서 이를 저장해둘 변수들 생성

(두 개의 트리를 병합하기 위해선 1개의 엣지가 무조건 추가되므로 추가될 엣지의 초기값을 1로 줌)

8. 엣지의 변경을 최소화하기 위해서 count1, count2를 비교해서 엣지의(노드의) 개수가 많은 트리를 먼저 생성함.

(i가 0일 경우는 따로 처리해서 0번 인데스에 저장되어있을 루트 노드를 먼저 생성)

(i가 0이 아닐 경우에는 return\_getParentNode()로 부모 노드 가져오고, insertDataBST()해서insertDataBST가 정상적으로 이루어졌을 경우에만(ret=1인 경우만) output 파일에return\_getParentNode()로 가져온 부모노드와 insert한 노드를 출력한다.)

(i가 0이 아닐 때 2\*i-1로 인덱스 값을 가져오는 이유는 1, 2, 3,,,일 때 배열에선 그럼 1, 3, 5,,인덱스의 값을 가져오게 된다. 근데 배열에는 엣지로 표현되어 있으니까 0, 2, 4,, 자리에는 부모 노드가 있는 것이니까 0, 2, 4,,,, 인덱스들은 넘기기 위해서)

9. 생성된 트리 밑에 남은 트리의 노드를 추가해준다. (i가 0일때는 따로 처리해서 루트노드 먼저 생성)

10. rm\_edges를 계산하기 위하여 return\_getParentNode()로 가져온 부모의 값과 insert될 노드가 저장되어 있는 배열 index에서 한 칸 앞에 저장되어있는(2\*i-2)값을 비교해서 다른 경우는 (원래의 트리에서 는 존재했던 엣지면서, T3에는 포함되지 않는 엣지) rm\_edges와 add\_edges 값을 1만큼 증가시켜준다.