



자료구조 (Data Structure)

Programming Assignment 6

HW 6

■ 문제 1:

2개의 텍스트 파일(input1.txt, input2.txt)로 **binary search tree**를 입력한다.

입력된 2개의 binary search tree가 각각
같은 위치에 같은 노드의 값을 가지고 있으면 YES를,
아니라면 NO를 출력하는 프로그램을 작성하라.

HW 6

예제	
입력 (input1.txt)	입력 (input2.txt)
3	3
10	10
12	12
9	9
출력	
YES	

HW 6

예제	
입력 (input1.txt)	입력 (input2.txt)
4	4
8	8
9	10
10	9
4	4
출력	
NO	

HW 6

◆ 입력(input1.txt):

- 첫 번째 줄은 읽어야 할 정수들의 개수 n
- 그 다음부터 n 줄에 양수인 정수가 한 줄에 하나씩 주어진다.

◆ 입력(input2.txt):

- 첫 번째 줄은 읽어야 할 정수들의 개수 n
- 그 다음부터 n 줄에 양수인 정수가 한 줄에 하나씩 주어진다.

◆ 출력:

- 각각 같은 위치에 같은 노드의 값을 가지고 있으면 YES 출력
- 각각 같은 위치에 같은 노드의 값을 가지고 있지 않으면 NO 출력

HW 6

- ◆ 제약 조건:
 - 입력은 file 입력, 출력은 stdout 출력
 - 노드의 값은 1 이상 50이하

HW 6

■ 문제 2:

Max heap의 insertion, deletion 함수를 작성하라.

각 노드는 data 값, parent node, left child node, right child node를 가지고 있다고 가정한다.

HW 6

예제	
입력	출력
i 4 i 4 i 5 d d d i 3 q	Insert 4 Exist number Insert 5 Delete 5 Delete 4 The heap is empty Insert 3

HW 6

◆ 입력:

- i k – heap에 key값이 k인 node를 insert한다.
- d – heap에서 가장 큰 key값을 가진 node를 delete한다.
- q – 프로그램을 종료한다.

◆ 출력:

- i k가 성공적으로 동작했을 경우 – **Insert k**
- i k가 실패한 경우(이미 존재하는 key값일 경우) – **Exist number**
- d가 성공적으로 동작했을 경우 **Delete K**
- d가 실패한 경우(heap이 비어있을 경우) – **The heap is empty**

HW 6

◆ 제약 조건:

- **Linked representation**을 사용할 것
- 주어진 input 형식 외에, **예외 input은 들어오지 않는다고 가정**
- 아래의 자료구조를 선언하여 사용할 것:

```
typedef struct node *treePointer;  
typedef struct node {  
    int key;  
    treePointer parent;  
    treePointer leftChild, rightChild;  
};
```

HW 6

■ 문제 3:

입력파일 **input.txt**로 주어진 n개의 양의 정수들을 읽어 **binary search tree**를 구성하고, 입력파일 **delete.txt**에 주어진 정수들을 구성된 **binary search tree**에서 삭제하는 프로그램을 작성하라.

삭제되어 재구성된 binary search tree는 **inorder와 preorder**로 출력하여 올바르게 트리가 구성되었는지 확인한다.

HW 6

예제	
입력 (input.txt)	입력 (delete.txt)
10 3 9 8 2 5 10 7 1 4 6	1 7 9 3 0

HW 6

예제

출력

inorder : 2 4 5 6 8 10
preoder : 2 8 5 4 6 10

HW 6

- ◆ 입력(input.txt) :
 - 첫 번째 줄은 읽어야 할 정수들의 개수 n
 - 그 다음부터 n 줄에 양수인 정수가 한 줄에 하나씩 주어진다.
- ◆ 입력(delete.txt) :
 - delete에 삭제할 정수들을 띄어쓰기로 구분에 한 줄에 입력한다.
 - 정수들이 주어진 입력의 끝은 0이다.
- ◆ 출력:
 - 삭제가 끝난 후 재구성된 tree의 inorder, preorder 출력

HW 6

◆ 제약 조건:

- 입력은 file 입력, 출력은 stdout 출력
- 노드의 값은 1이상 50이하
- 삭제할 node의 left, right 노드가 둘 다 존재할 시, left subtree에서 가장 큰 값으로 replace한다.
- delete.txt에 입력할 삭제할 노드의 값은 input.txt에 존재하는 노드의 값만 입력

제출 방법

■ 소스코드:

- ◆ 파일 이름: HW6_학번_문제번호.c(or .cpp)
ex)HW6_20220000_1.c(or .cpp)
- ◆ 확장자는 **무조건** .c 혹은 .cpp 이어야 함. (입출력 파일 이외의 프로그램 파일)
이외의 파일(.txt 등)은 **절대** 받지 않음(미제출로 간주)
- ◆ 컴파일 에러가 발생할 경우 0점 처리
- ◆ 무한 루프 / 세그멘테이션 오류는 해당 testcase 0점 처리
- ◆ 입출력 양식이 틀릴 경우 감점

제출 방법

■ 보고서:

◆ 파일 이름: HW6_학번_Document.pdf

◆ 반드시 PDF 파일로 제출할 것

이외의 파일(.docx, hwp 등)은 절대 받지 않음(미제출로 간주)

제출 방법

■ 압축 파일:

◆ 이름: HW6_학번.zip

ex) HW6_20220000.zip

◆ 압축을 풀면 아래의 파일들이 있어야 함:

HW6_학번_1.c(or .cpp) && input1.txt, input2.txt

HW6_학번_2.c(or .cpp)

HW6_학번_3.c(or .cpp) && input.txt, delete.txt

◆ HW6_학번_Document.pdf

제출 형식이 틀릴 경우 과제 점수의 30% 감점

제출 방법

- ◆ 6/12 24:00(자정)까지 **(메일 발송 시간 기준)**
- ◆ sguds.yj@gmail.com 으로 압축파일(HW6_학번.zip) 제출
- ◆ 제출 기한 이후의 메일은 **미제출로 간주함**
- ◆ 과제 채점은 **cs pro** 기준
- ◆ Copy 검사 실시