

## Lab 06 (4/7) Binary Search Tree

**Due : ~2021.04.12.(Tue) 23:59, Late Submission : ~2021.04.13.(Wed) 23:59**

- **Binary Search Tree Implementation - Insert, Delete, Find, PrintInorder.**
- 아래의 사진과 같이 [input].txt 파일을 입력받아 BST를 생성하고 편집하는 기능을 구현하고, 각 커맨드에 알맞은 결과 값을 [output].txt 파일에 출력하여 저장한다.
- **Implement the function of creating and editing a BST by receiving the [input].txt file as input as below and the output result of input commands is stored in the [output].txt file.**

```

열기(O)▼ 파일 input1.txt /data/data_ba... 저장(S) 열기(O)▼ 파일 output1.txt /data/data_b... 저장(S)
i 5
i 1
i 10
pi
f 5
d 5
pi
insert 5
insert 1
insert 10
1 5 10
5 is in the tree
delete 5
1 10

```

- <input> :
   
각 line마다 Command가 주어짐. (**insert, delete, print, find**)
   
**Commands are given line by line (insert, delete, print, find)**
- <output> :
   
각 **Command**에 맞는 **result** 출력.
   
**Appropriate output messages for each command.**

### <Insert>

- 위의 사진에서처럼 "i [insert할 key값]"로 표현합니다.
- "i 5"의 경우 "5"노드를 BST에 추가하시고, "insert 5" 메시지를 출력시켜주시면 됩니다.
- Tree 안에 이미 존재하는 key값을 insert하는 경우 에러메시지를 출력해주셔야 합니다.
   
예러메시지 형식은 "insertion error: key값 is already in the tree"입니다.
- "i [key value to be inserted]"
- In the case of "i 5", node "5" should be inserted into BST, and the message "insert 5" should be printed out.
- When inserting the key value which already exists in BST, the error message should be printed out, in the format of "insertion error: key Value is already in the tree".

### <Delete>

- 위의 사진에서처럼 "d [delete할 key값]"로 표현합니다.
- "d 5"의 경우, Tree 안에 존재하는 "5"노드를 제거하시고, "delete 5" 메시지를 출력시켜 주시면 됩니다.
- Tree안에 존재하지 않는 key값을 제거하려 할 경우 에러 메시지를 출력해주셔야 합니다.
   
예러메시지 형식은 "deletion error: key값 is not in the tree"입니다.
- "d [key value to be deleted]"

- In the case of “d 5”, node “5” should be deleted from BST, and the message “delete 5” should be printed out.
- When deleting the key value which does not exist in BST, the error message should be printed out, in the format of “**deletion error: key value is not in the tree**”.

#### <Find>

- 위의 사진처럼 “f [search할 key값]”로 표현합니다.
- key값이 존재하는 경우, “**key값 is in the tree**”
- key값이 존재하지 않는 경우, “**finding error: key값 is not in the tree**”로 출력해주세요 합니다.
- “f [**key value to be searched**]”
- When the key value exists in the tree, “**key value is in the tree**” ,
- when the key value does not exist in the tree, “**finding error: key value is not in the tree**” should be printed out.

#### <PrintInorder>

- 위 사진처럼 “pi”로 표현 합니다.
- “pi”的 경우, Tree 전체를 Inorder로 순회하며 출력해주시면 됩니다.
- 출력 형식은 “**key값 key값 key값 ...**”입니다. (마지막 숫자 이후 공백없이 줄바꿈.)
- Tree가 비어있을 경우, “**tree is empty**”를 출력해주시면 됩니다.
- “pi”
- In the case of “pi”, the entire tree should be printed inorder.
- Printing format is “**key값 key값 key값 ...**”, all the spaces are one space.
- If the tree is empty, “**tree is empty**” should be printed out.

#### <Structure & Function Format>

##### Structure

```
typedef struct BST* Tree;
typedef struct BST{
    int value;
    struct BST* left;
    struct BST* right;
}BST;
```

##### Function

```
Tree insertNode(Tree root, int key);
Tree deleteNode(Tree root, int key);
int findNode(Tree root, int key);
void printInorder(Tree root);
void deleteTree(Tree root);
```

- 위 사진과 같은 Struct 구조체를 사용하셔야 합니다.
- 위 사진과 같은 함수들을 형식에 맞게 구현해주시면 됩니다.
- Struct format above should be used for implementation
- Functions should be implemented in appropriate format as above.

#### <File Name Format>

- [StudentID].c      ex) 20XXXXXXXXXX.c

## <Execution>

- gcc 20XXXXXXXXXX.c -o 20XXXXXXXXXX
  - ./20XXXXXXXXXX [input\_file\_name] [output\_file\_name]
  - !!! 꼭 제공되는 testcase로 실행시켜보시기 바랍니다!!!!,
  - !!! Run your solution code with the provided test case above and check whether it works properly !!!

## <Issue>

- 코드 작성시 주석을 적어주시기 바랍니다. 주석이 없는경우 Cheating으로 간주될 수 있습니다.
  - 제공된 testcase는 채점 case에 포함됩니다. 모두 알맞게 나오는지 확인해보시기 바랍니다.
  - 파일 입출력은 `argv[]`를 사용하여 구현해주시기 바랍니다.
  - 제출 마감 시간 이전의 가장 최신 버전의 commit을 기준으로 채점할 예정입니다.
  - 제출 파일과, 풀더 naming 은 꼭 지정된 형식으로 해주셔야 합니다.
  - Please write down the detailed comments when writing the code. If there is no comment, it might be considered cheating.
  - Provided test case is included in the test case for grading. Please check to see if it makes a proper result.
  - Do not use a fixed file name when inputting and outputting files, but implement it using `argv[]` as in skeleton code.
  - Scoring will be based on the latest version of commit before the deadline.
  - The names of the .c file and directory should be named in proper format.  
  - 출력시, 위 사진의 예시와 같은 형식으로 출력해주시면 됩니다. 모든 공백은 띄어쓰기 한칸입니다. 모든 출력 메시지의 알파벳은 소문자만 사용하여 출력합니다.
    - > 출력시 위 사진과 같이 커맨드에 알맞는 메시지 출력후 줄바꿈(\n).
    - > 출력파일 마지막줄의 경우, 줄바꿈(\n) 후 EOF.
  - All the messages must be printed out according to the appropriate format as shown in the example above. All spaces are one space. Only lowercase letters should be used for the alphabets in output messages.
    - > newline(\n) after each output message.
    - > newline(\n) after the last output message.

## <Directory Format>

- 아래와 같이 git 프로젝트 폴더에 "lab06" 폴더 생성후, "lab06" 폴더 안에 "20XXXXXXXXXX.c" 파일을 위치시키시면 됩니다.
  - After creating the "lab06" directory in the git-project-directory as below, place the "20XXXXXXXXXX.c" file in the "lab06" directory

For a detailed analysis see the tables and figures in the main report.

2022\_CSE2010\_20xxxxxxx/ (GitLab project directory)

1

---lab05/

-----2020XXXXXX.c

— 1 —

--lab06/

-----2020XXXXXX.c

>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> QnA : 2021datastructure@gmail.com