



자료구조 (Data Structure)

Programming Assignment 4

(Ch5에 대한 Homework)

HW4

- 문제 1:

Write insertion and deletion functions for a max heap represented as a linked binary tree. Assume that each node has a parent field as well as the usual left child, right child, and data fields.

HW4

- 문제 1:

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

HW4

- 문제 1:
 - 입력:
 - i k - heap에 key값이 k인 node를 insert한다.
 - d - heap에서 가장 큰 key를 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**

HW4

- 문제 1:
 - 제약 조건:
 - Linked representation을 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;  
};
```

HW4

- 문제 2:

Write a C program to construct binary search tree from given preorder traversal, and perform inorder and postorder traversal on it.

HW4

- 문제 2:

예제	
입력	출력
6 30 5 2 40 35 80	Inorder: 2 5 30 35 40 80 Postorder: 2 5 35 80 40 30
6 30 5 5 40 35 80	cannot construct BST

HW4

- 문제 2:
 - 입력:
 - 첫 번째 줄 - tree node의 개수 n
 - 두 번째 줄 - n 개의 key value
 - 출력:
 - 한 줄에 하나씩, inorder traversal과 postorder traversal에 대한 결과
 - 중복되는 key가 있을 경우, “cannot construct BST” 출력

HW4

- 문제 2:
 - 제약 조건:
 - 반드시 tree를 만든 후, inorder와 postorder traversal을 진행할 것
 - tree를 만들지 않은 경우, 0점 처리

HW4

- 문제 3:

Write a C functions for a max priority queue that represents the priority queue as a binary search tree. Your codes for *top*, *pop* and *push* should have complexity $O(h)$, where h is the height of the search tree.

HW4

- 문제 3:

예제	
입력	출력
push 3	Push 3
push 3	Exist number
top	The top is 3
push 5	Push 5
top	The top is 5
pop	Pop 5
push 3	Exist number
pop	Pop 3
pop	The queue is empty
q	

HW4

- 문제 3:
 - 입력:
 - **push k** - queue에 key값이 k인 node를 insert한다.
 - **top** - heap에서 가장 큰 key를 가진 node를 출력한다.
 - **pop** - heap에서 가장 큰 key를 가진 node를 delete한다.
 - **q** - 프로그램을 종료한다.
 - 출력:
 - push k가 성공적으로 동작했을 경우 - **Push k**
 - push k가 실패한 경우 (이미 존재하는 key값일 경우) - **Exist number**
 - top이 성공적으로 동작했을 경우 - **The top is k**
 - pop이 성공적으로 동작했을 경우 - **Pop k**
 - top/pop이 실패한 경우 (queue가 비어있을 경우) - **The queue is empty**

HW4

- 문제 3:
 - 제약 조건:
 - Binary search tree를 사용할 것
 - 주어진 input 형식 외에, 예외 input은 들어오지 않는다고 가정

제출 방법

■ 소스코드

◆ 이름: **HW4_학번_문제번호.c**

● ex) HW4_20187777_1.c

◆ cpp, txt 파일 등은 **절대** 받지 않음, 반드시 .c 파일로 제출할 것

◆ 컴파일 에러가 발생할 경우 0점 처리

◆ 무한 루프/세그멘테이션 등의 오류는 해당 testcase 0점 처리

◆ 입출력 양식이 틀릴 경우 감점

제출 방법

■ 보고서

- ◆ 이름: [보고서.pdf](#)
- ◆ 반드시 pdf 파일로 변환 후 제출
- ◆ 조교가 올린 report 양식을 참고하여 작성 (이번에도 시간복잡도 생략)
- ◆ **hard copy 제출을 하지 않으니 제출 양식이 다르다면, 무조건 0점 처리**
- ◆ **추후 대면 수업으로 전환 시 제출 방법 변경 사항 공지 예정**

제출 방법

■ 압축 파일

◆ 이름: **HW4_학번.zip**

● ex) HW4_20187777.zip

◆ 압축을 풀면 아래 파일들이 있어야 함.

● HW4_학번_1.c

● HW4_학번_2.c

● HW4_학번_3.c

● HW4_보고서.pdf

◆ 제출 형식이 틀릴 경우 과제 점수의 50% 감점

제출 방법

- 12/03 저녁 12:00까지 tyeun7@u.sogang.ac.kr 으로 제출
압축파일(HW4_학번.zip) 제출
- 과제 채점은 gcc compiler 기준. cspro에서 컴파일 안되면 그냥 0점
- 명시된 제출 마감 시간보다 늦는 것은 절대 받지 않음
- Random copy 검사