

Lab 8

Max heap

2019. 04. 25

lab 8. Max heap

- **Data Structure Specification**

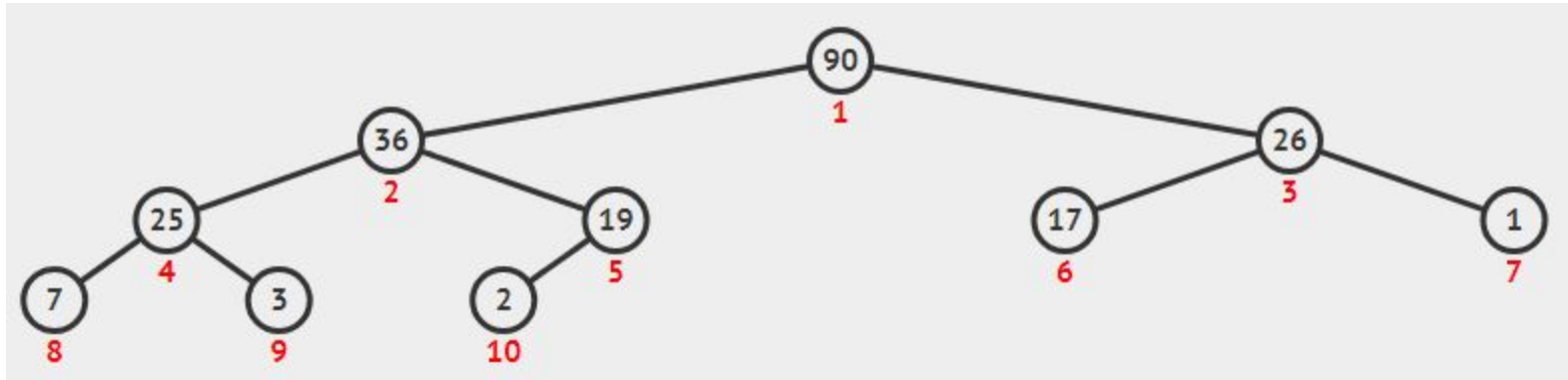
```
struct HeapStruct
{
    int Capacity;
    int Size;
    ElementType *Elements;
}
```

- **Function specification**

- void Insert(ElementType X, PriorityQueue H)
 - insert X and print 'insert X'
 - if X already exists, then print 'X is already in the tree'.
 - if heap is full, then print 'heap is full'
- void Find(ElementType X, PriorityQueue H)
 - if X is in the heap, print 'X is in the heap'.
 - if X is not in the heap, print 'X is not in the heap'.
- void Print(PriorityQueue H)
 - Print all Elements in index order.

lab 8. Max heap

- void Print(PriorityQueue H)



90 36 26 25 19 17 17 3 2

lab 8. Max heap

- **Input**

- The first line contains size of heap
- **i x** : insert element x
- **f x** : find element x
- **p** : print the entire max heap in index order.

- **You have to use file I/O like the previous assignment.**

input - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
5
i 4
i 1
i 9
i 10
i 1
f 10
f 5
p
```

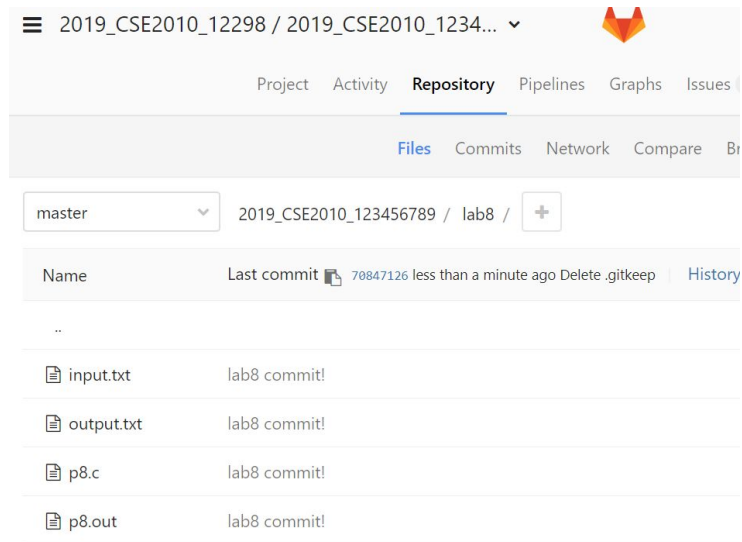
output - 메모장

파일(F) 편집(E) 서식(O) 보기(V)

```
insert 4
insert 1
insert 9
insert 10
1 is already in the heap.
10 is in the heap.
5 is not in the heap.
10 9 4 1
```

lab 8. Max heap

- Submission
 - Project directory name : lab8
 - Source file name : p8.c
 - Executable file name : p8.out
 - You should upload in the hconnect (Gitlab) server.



DeadLine

Wednesday, 8 May, 23 : 59 pm