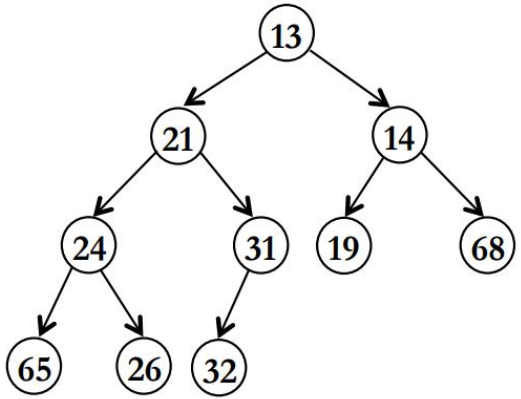
Lab 08: Max Heap

Data Structure 2023

Binary(Min) Heap

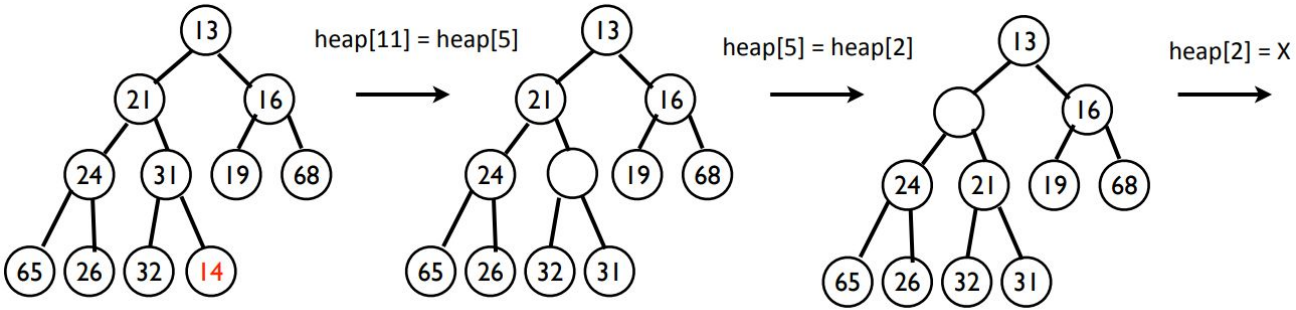
- A min heap is a **complete binary** tree and **partially ordered tree** in which the key value in each node is no larger than the key values in its children
- **Complete tree**
  - Every level of tree is completely filled, with the exception of the bottom level, which is filled from left to right



Min Heap - Insertion

insertion of 14

x=14



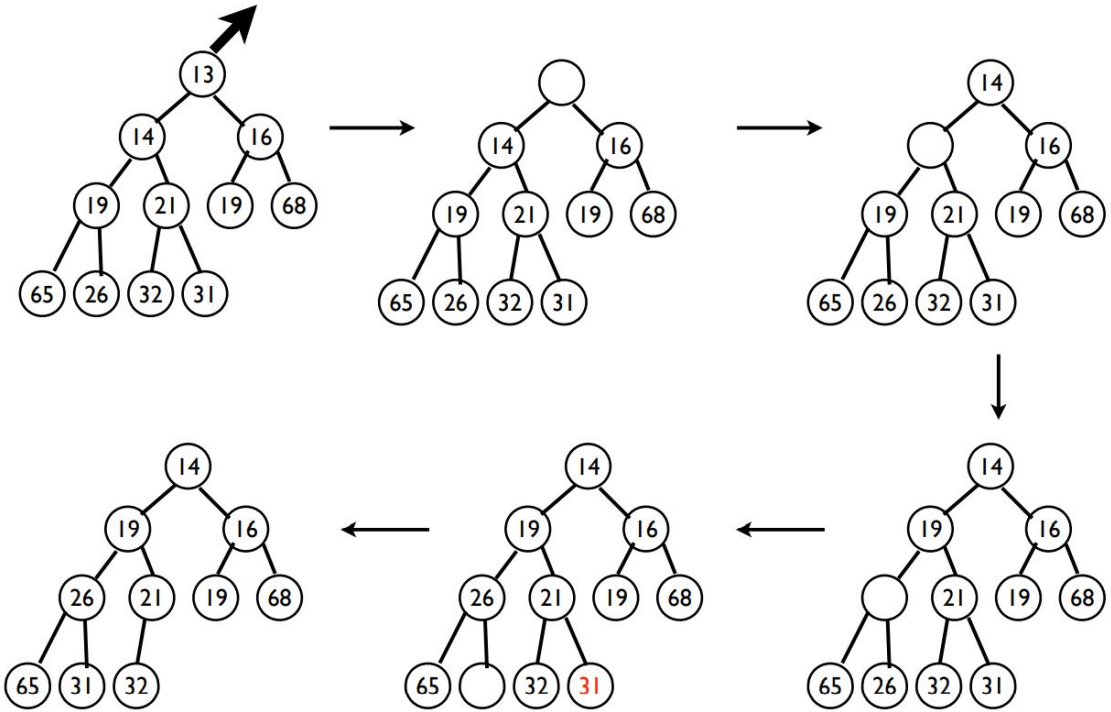
i=11
heap[floor(i/2)] ≤ x ?

i=5
heap[floor(i/2)] ≤ x ?

i=2
heap[floor(i/2)] ≤ x ?

	13	21	16	24	31	19	68	65	26	32	14
i	0	1	2	3	4	5	6	7	8	9	10	11

Min Heap - Deletion



Max Heap ADT

- **Heap* CreateHeap(int heapSize)**
  - Create a heap with the given size.
- **int Insert(Heap *heap, int value)**
  - Insert a new key to the max heap. You should find the right position for the new key to maintain the max heap. Return the result(0: success, 1: full, 2: duplicated).
- **int Find(Heap *heap, int value)**
  - Find the key in the heap. Return 1 if the value exists. Otherwise, return 0.
- **int DeleteMax(Heap *heap)**
  - Find Delete the max in root node and reconstruct the heap to maintain max heap. Return what node you have deleted. If the heap is empty, return 0.
- **int* GetElements(Heap *heap)**
  - Return an array containing all elements from the heap in level order traversal.

Max Heap ADT

- **int IsFull(Heap *heap)**
  - Check if the heap is full. Return 1 if heap is full; otherwise, return 0.
- **int IsEmpty(Heap *heap)**
  - Check if the heap is empty. Return 1 if heap is empty; otherwise, return 0.
- **int Size(Heap *heap)**
  - Return the size of the heap.

Max Heap ADT

- **n x**
  - Create a new heap with the size of x. The number x is the maximum size of the MaxHeap. This operation will always be given in the first line of the operations in your input file.
- **i x**
  - Insert a new key “x” into the max heap. **Print what key you inserted.**
- **f x**
  - Find the given key to check whether the key exists in the heap, and **print whether the key exists or not.**
- **d**
  - Delete the max key in the root node. **Print what node you have deleted.**
- **p**
  - **Print the entire max heap in level order traversal.**

Max Heap ADT

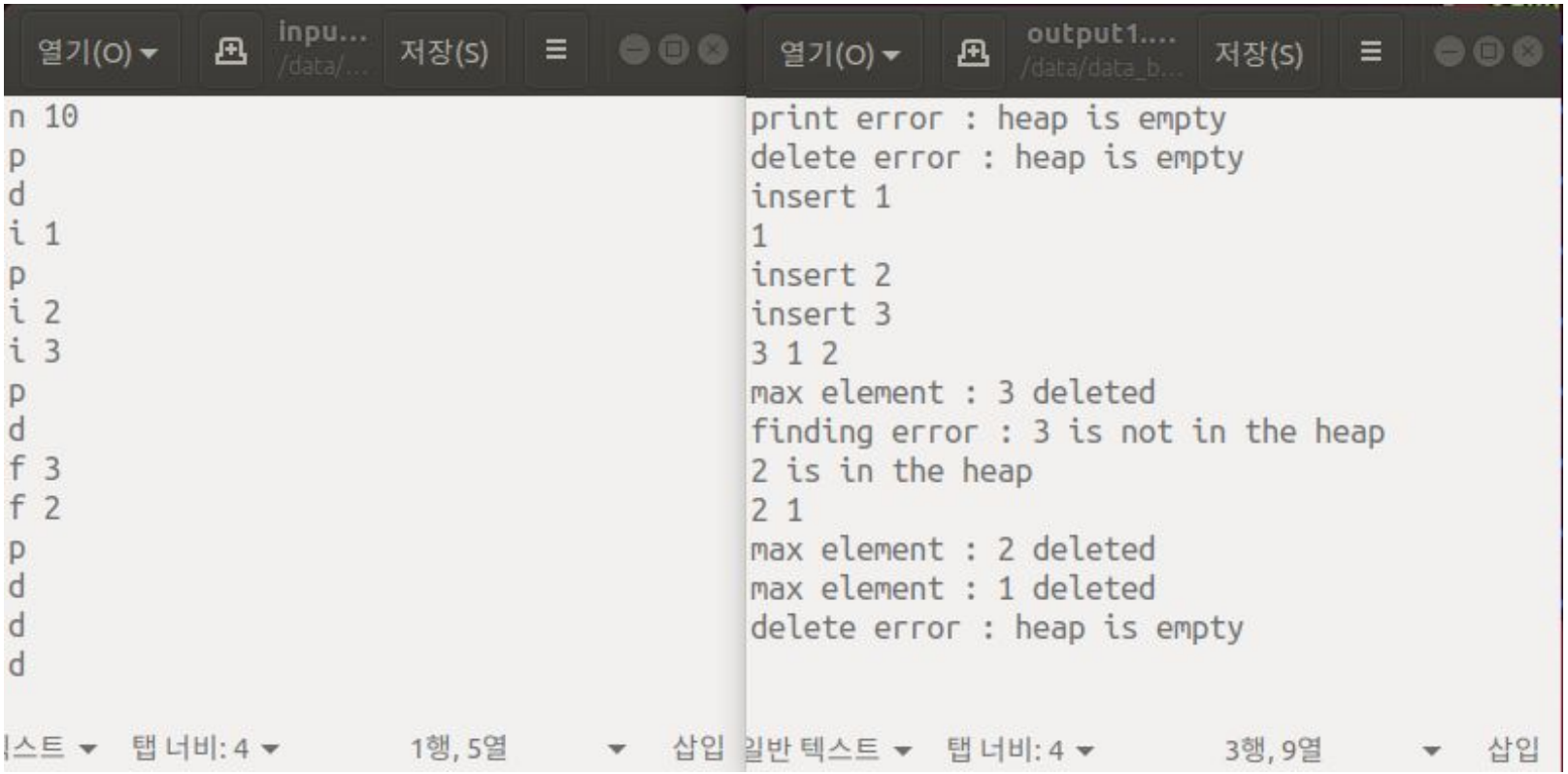
Structure

```
typedef struct HeapStruct{
    int Capacity;
    int Size;
    int *Elements;
}Heap;
```

Function

```
Heap* CreateHeap(int heapSize);
int Insert(Heap *heap, int value);
int Find(Heap *heap, int value);
int DeleteMax(Heap* heap);
int* getElements(Heap* heap);
int IsFull(Heap *heap);
int IsEmpty(Heap *heap);
int Size(Heap *heap);
```

Input & Output Example



Assignment

- Due
  - **~ 2023.05.10(수) 23:59**
  - Last Commit 기준

- 자세한 내용은 과제 명세 PDF 파일 참고