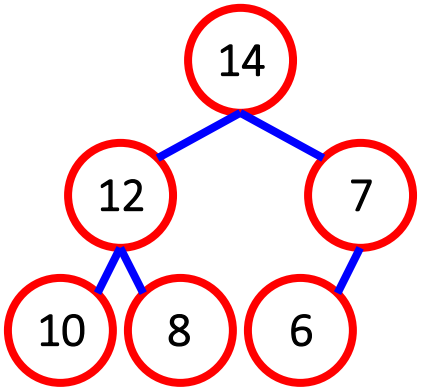
# CS235101
# Data Structure
# Homework 3

Due date: 2017/11/23 23:59

# Max Heap

- A ***max tree*** is a tree in which the key value in each node is ***no smaller*** than the key values in its children. A ***max heap*** is a ***complete binary tree*** that is also a ***max tree***.

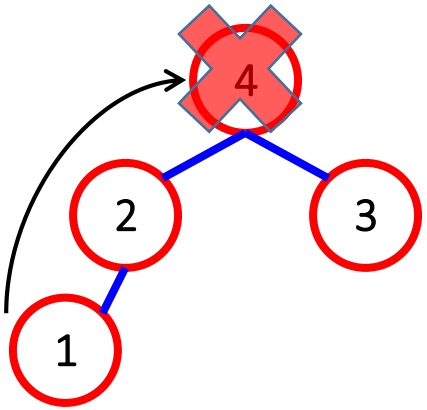


# Target

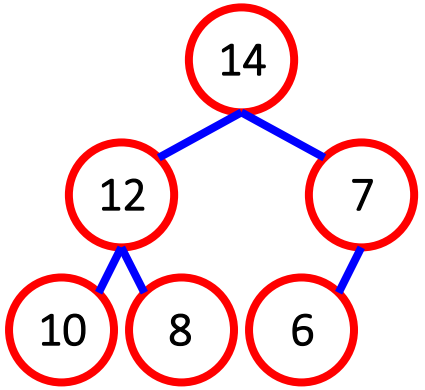
- The target of this homework is to implement a max heap
- You are asked to implement 6 functions below
  - Insert (int value)
  - DeleteMax()
  - MaxPathWeight (int index)
  - InorderTraversal (int index)
  - PreorderTraversal (int index)
  - PostorderTraversal (int index)

# Target

- Insert (int value)
  - Insert a data into the heap
- DeleteMax()
  - Delete the root node.

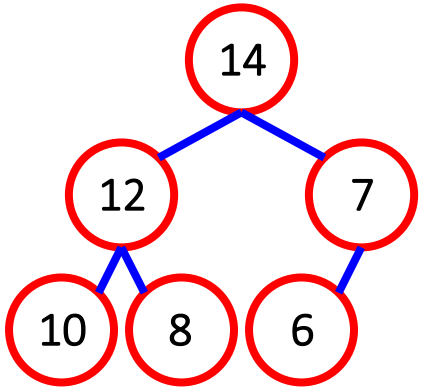


# Target




- MaxPathWeight (int index)
  - Return the max path weight from root to leaf.
  - Ex: 14+12+10 is the maximum path weight, so output 36
- InorderTraversal (int index)
  - Return the inorder traversal path from root.
  - Ex: 10 12 8 14 6 7

# Target




- PreorderTraversal (int index)
  - Return the preorder traversal path from root.
  - Ex: 14 12 10 8 7 6
- PostorderTraversal (int index)
  - Return the postorder traversal path from root.
  - Ex: 10 8 12 6 7 14

Each traversal path output format:

- Int Int Int Int …… Int

# File Structure

- #include "function.h"
- **class** "Heap" represents a Heap
  - size : Use this variable to record your heap size.
  - heap: Use this array to construct the heap.
  - In the end, we use above variable to print the whole heap.
- **class** "Implement" contains your implementation
- Sample input and output

# Heap

```
//It contains the functions that you have to override.
class Heap
{
public:
    int size=0;
    int *heap=new int[100000];

    virtual void Insert(int value)=0;
    virtual void DeleteMax()=0;
    virtual int MaxPathWeight(int index)=0;
    virtual string InorderTraversal(int index)=0;
    virtual string PreorderTraversal(int index)=0;
    virtual string PostorderTraversal(int index)=0;
```

# Implement

- In function.h

```
class Implement : public Heap
{
public:
    void Insert(int value);
    void DeleteMax();
    int MaxPathWeight(int index);
    string InorderTraversal(int index);
    string PreorderTraversal(int index);
    string PostorderTraversal(int index);
};
```

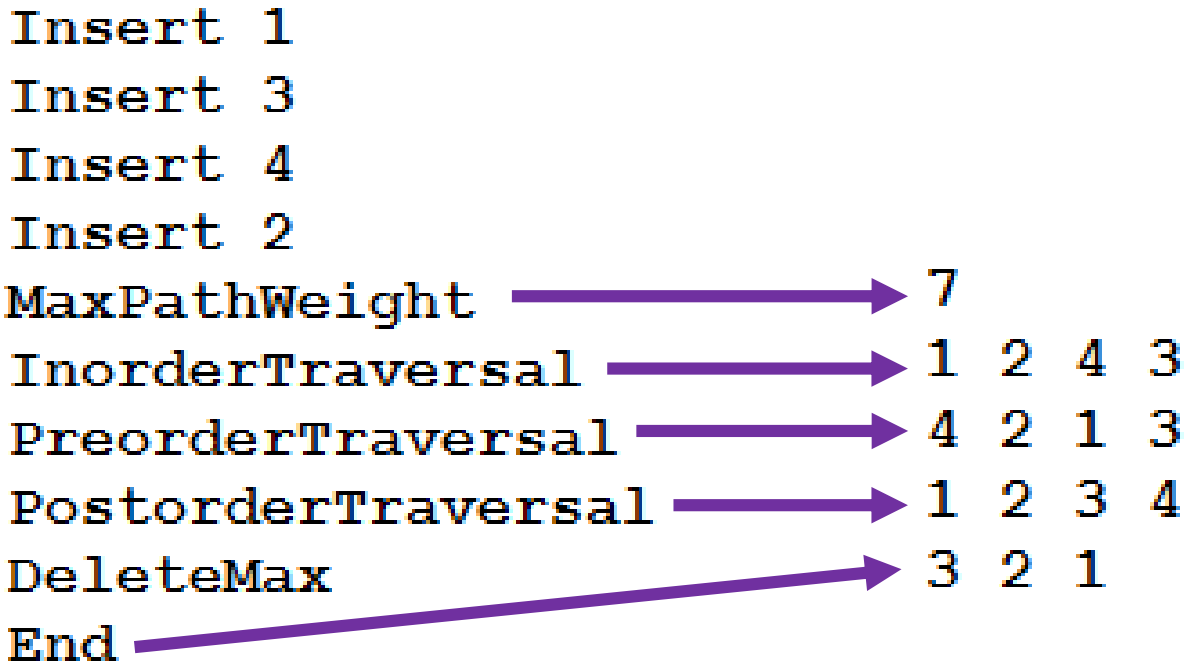
- code.cpp

```
#include "function.h"

using namespace std;

//add your code here
//--------------------------------------
void Implement::Insert(int value)
{
```

# Sample IO




**End will output the whole heap.**
**But you don't need to handle it.**

# Note

1.Each data is an integer ranges from 1 to 99999

2.Each node is unique, two duplicate integers won’t exist at the same time

3.There will exists at least 1 and at most 99999 nodes in the final heap.

4.Root index is 1.

# STL is not allowed

- <list> <vector> <forward_list> … are not allowed
- If you try to include the above headers, your source files WILL NOT be compiled properly during TA's evaluation

# Submission

- Online Judge: #11653
- Archive your source codes (whole hw3 folder) into a zip file named [studentID]_hw3.zip
  - E.g. 102062999_hw3.zip
- Submit the zip file to ilms system BEFORE the deadline