Programming Assignment #4: Order Management System

(due 23:59, Dec 25th, 2020)

Objective:

In this assignment, you need to write a C++ program to manage the orders in a company, including adding orders, deleting orders, and searching orders. Every order contains its **id** and the time information, **date**, within that the order can be finished. Build up a **Tree** according to the **date** of each order.

Provided files:

(1) ***main.cpp***:

Parse the given input file, execute your functions, and check your answers.

(2) ***OrderMGMT.h***:

I. A node structure which you **cannot** change:

```
unsigned id // every order has its unique id
unsigned date // time information
unsigned leftSize // store the size of left subtree
Node *left // left subtree
Node *right // right subtree
```

II. An OrderMGMT class including private and public members and functions that you can change.

(3) ***OrderMGMT.cpp***:

Include 4 functions to be implemented by you.

void OrderMGMT::addOrder(unsigned date, unsigned id)

Add one order to your order management system according to its **date**.

Constrains: If the **date** of the new order is the same as the **date** of the order already in your system, you **cannot** take the order.

void OrderMGMT::deleteOrders(unsigned start, unsigned end)

Delete orders whose **date** is within the time interval defined from **start** to **end**.

Constrains: The time interval is a closed interval.

list<unsigned> OrderMGMT::searchByDate(unsigned start, unsigned end)

Search your tree to find the orders whose **date** is within the time interval defined from **start** to **end**. Store their **id** in a list.

Constrains: The time interval is a closed interval.

The **id** in the list are sorted according to their **date** (earliest first).

list<unsigned> OrderMGMT::searchByDateRank(unsigned a_th, unsigned b_th)

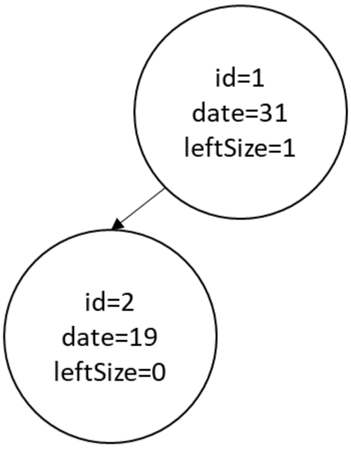
Search your tree to find a sequence of orders starting from the **a_th** rank of date and ending with the **b_th** rank of **date**. Store their **id** in a list.

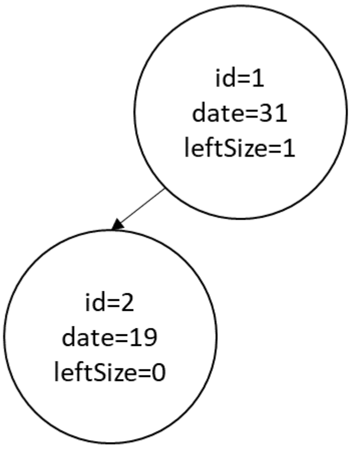
Constrains: The **id** in the list are sorted according to the rank of their **date** (earliest first).

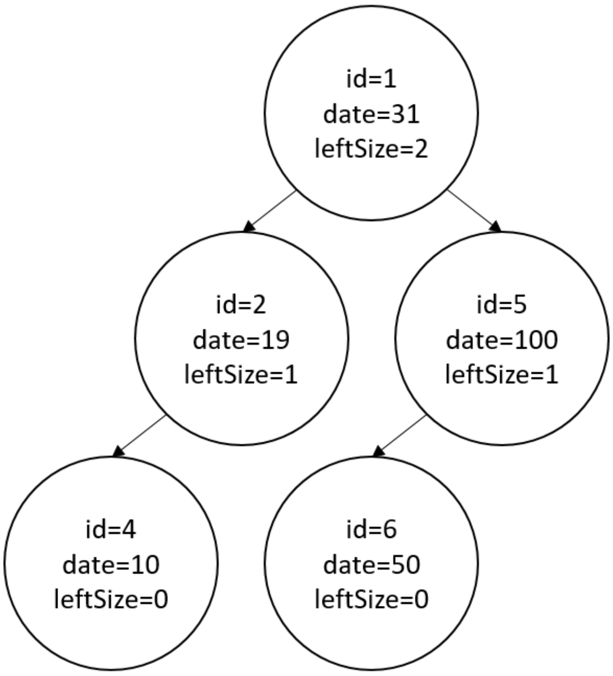
The order with the earliest **date** ranks 1st.
If **b_th** is larger than the number of the total orders, store the order sequence from the **a_th** rank of **date** to the number of the total orders.
If **a_th** is larger than the number of the total orders, store nothing.

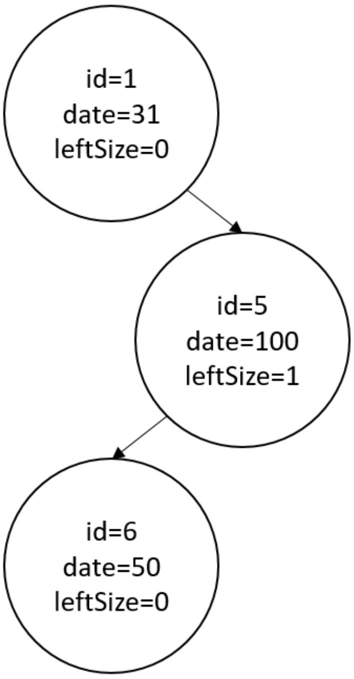
(4) ***TestCase***:

Input	Example tree or returned **list**

addOrder 31 1
addOrder 19 2
	 id=1
	date=31
	leftSize=1
	id=2
	date=19
	leftSize=0

addOrder 31 3	 id=1
	date=31
	leftSize=1
	id=2
	date=19
	leftSize=0

addOrder 10 4
addOrder 100 5
addOrder 50 6
	 id=1
	date=31
	leftSize=2
	id=2
	date=19
	leftSize=1
	id=5
	date=100
	leftSize=1
	id=4
	date=10
	leftSize=0
	id=6
	date=50
	leftSize=0

deleteOrders 10 20	 id=1
	date=31
	leftSize=0
	id=5
	date=100
	leftSize=1
	id=6
	date=50
	leftSize=0

searchByDate 10 50	1 6
searchByDateRank 1 3	1 6 5
searchByDateRank 2 5	6 5
searchByDateRank 4 5	

Language:

C++.

Platform:

You may develop your software on UNIX/Linux.

Compile: $ g++ main.cpp -o hw4

Execution: $ ./hw4 <input file>

Submission

Please compress the following files into a zip file and name it by your ***student ID***. For example, **"HW4_0850232.zip"**. Then upload the compressed file to the new E3 website by the deadline (Dec 25th,2020).

(1) OrderMGMT.h

(2) OrderMGMT.cpp

Grading policy:

(1) Example case correctness (60%)

(2) Hidden case correctness (10%)

(3) Hidden case ranking (30%) (ranking priority: accuracy > run time)

(4) The runtime limit of this homework is 6hr. If the runtime exceeds 6hr, you will fail this homework.