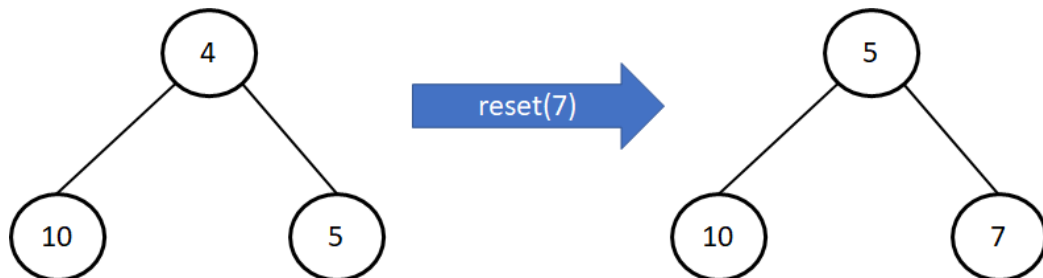


การจำลองการจัดการ Bandwidth Management

ในการส่ง data package ในเครือข่ายการส่งข้อมูลจากที่หนึ่งไปยังอีกที่หนึ่ง ในบางกรณีนั้นจะมีการกำหนดความสำคัญของ package ไปด้วย เมื่อ router จะทำการส่งข้อมูลจะพิจารณาความสำคัญนี้ด้วยโดยจะส่งข้อมูลที่มีความสำคัญสูงก่อน ในกรณีนี้เราจะให้ค่าน้อยถือว่ามีความสำคัญที่สุดและเก็บด้วยโครงสร้าง Heap

หน้าที่ของเราคือเขียน Method จำลองการจัดการ bandwidth ของ router ดังต่อไปนี้

- **int min()** เป็น method ที่คืนค่าความสำคัญของ data package ที่น้อยที่สุด
- **int removeMin()** เป็น method ที่คืนค่าความสำคัญของ data package ที่น้อยที่สุดและนำเอา data package นั้นออก (นั่นคือเอาค่าความสำคัญออก)
- **void insert(int k)** เป็น method ในการเพิ่ม data package โดยจะเก็บค่าความสำคัญ
- **void downHeap()** เป็น method ที่ปรับโครงสร้าง tree ให้มีลักษณะเป็น min Heap อย่างไรก็ตาม router มีฟังก์ชันพิเศษคือ ปรับค่าความสำคัญน้อยสุดเป็นค่าอื่น (เหมือนกับว่าเราสามารถรีเซ็ตค่าความสำคัญได้นั่นเอง)
- **void reset(int k)** เป็น method ที่เปลี่ยนค่าความสำคัญที่น้อยที่สุดเป็นค่า k เมื่อเรียกใช้งานแล้วปรับโครงสร้างของ tree ให้ถูกต้องด้วย ตัวอย่างเช่น หากเริ่มต้นมีข้อมูลใน Heap เป็นรูป tree ต้นซ้ายมือ หลังจากเรียก เมื่อเรียก reset(7) แล้วผลลัพธ์จะเป็นดัง tree ต้นขวามือ



ขั้นตอนการทำงานบ้าน

1. ดาวน์โหลดไฟล์การบ้านได้ที่ <https://github.com/CS-CMU/cs252student>
2. ในโฟลเดอร์ HW07 จะมีไฟล์ main.cpp และ HW07.cpp
3. ไฟล์ main.cpp จะเป็นไฟล์สำหรับทดสอบโปรแกรมของนักศึกษา สามารถแก้ไขได้ตามความเหมาะสม ไฟล์นี้ไม่ต้องส่งและไม่มีการตรวจ
4. ไฟล์ HW07.cpp จะเป็นไฟล์ที่นักศึกษาต้องแก้ไข โดยมีรายละเอียดดังนี้

คลาส Router ประกอบไปด้วย

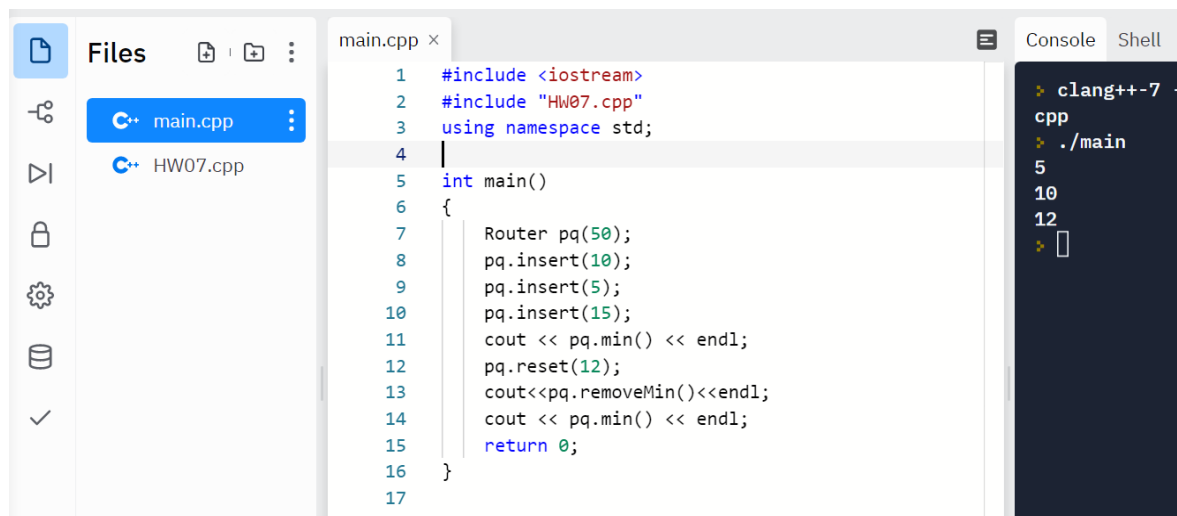
1. ตัวแปร Pointer `int *data`; ตัวแปรตัวนี้ต้องประกาศเป็น `public` เท่านั้นทำหน้าที่ในการเก็บข้อมูลความสำคัญ (ใช้สำหรับการตรวจ)
2. Constructor สำหรับคลาส Router ให้ค่าเริ่มต้นกับ ขนาดของ tree
3. ฟังก์ชัน `int removeMin()` คืนค่าความสำคัญของ package ที่น้อยที่สุดพร้อมทั้งนำความสำคัญ ออก
4. ฟังก์ชัน `int min()` คืนค่าความสำคัญของ package ที่น้อยที่สุด
5. ฟังก์ชัน `void insert (int k)` นำค่าความสำคัญของ package เข้าไปเป็นใน tree
6. ฟังก์ชัน `void downHeap(int k)` ปรับโครงสร้าง tree เมื่อพิจารณาโหนดที่มีความสำคัญ `i`
7. ฟังก์ชัน `void reset(int k)` ปรับข้อมูลความสำคัญที่น้อยที่สุดเป็น `k` แล้วปรับโครงสร้างของ tree ให้ถูกต้อง

ฟังก์ชันที่นักศึกษาต้องเขียนส่งคือ `Router()`, `removeMin()`, `min()`, `insert()`, `downHeap()` และ `reset()`

นักศึกษาสามารถสร้างตัวแปรหรือฟังก์ชันเพิ่มได้ตามความเหมาะสม (ไม่ตรวจ)

5. ส่งไฟล์ HW07.cpp ที่ <https://gdr252.cs.science.cmu.ac.th> คะแนนที่ได้ในเว็บเกรดเดอร์คือคะแนนที่นำไปใช้ตัดเกรด

ตัวอย่าง



```
1 #include <iostream>
2 #include "HW07.cpp"
3 using namespace std;
4
5 int main()
6 {
7     Router pq(50);
8     pq.insert(10);
9     pq.insert(5);
10    pq.insert(15);
11    cout << pq.min() << endl;
12    pq.reset(12);
13    cout<<pq.removeMin()<<endl;
14    cout << pq.min() << endl;
15    return 0;
16 }
17
```

```
> clang++-7 -
cpp
> ./main
5
10
12
> []
```