

เลขที่นั่งสอบ

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ข้อสอบปลายภาคการศึกษาที่ 2/2556

วันจันทร์ที่ 12 พฤษภาคม 2557

เวลา 9.00 - 12.00 น.

วิชา CPE 113 Algorithms and Data Structures.

น.ศ. วิศวกรรมคอมพิวเตอร์ชั้นปีที่ 1A,B

คำสั่ง

1. ข้อสอบมีทั้งสิ้น 6 ข้อ จำนวน 8 แผ่น(รวมแผ่นนี้) ข้อละ 5 คะแนน คิดเป็นคะแนนรวมทั้งสิ้น 30 คะแนน
2. ทำทุกข้อลงในข้อสอบที่เว้นช่องไว้ให้
3. ไม่อนุญาตให้นำเครื่องคำนวณใดๆเข้าห้องสอบ
4. ห้ามนำเอกสารใดๆ เข้าห้องสอบ
5. เขียนชื่อ และ รหัสประจำตัว ลงในกระดาษคำตอบทุกแผ่น

.....
(อ.พิพัฒน์ ศุภศิริสันต์)
ผู้ออกข้อสอบ

ข้อสอบนี้ได้ผ่านการประเมินจากภาควิชาวิศวกรรมคอมพิวเตอร์แล้ว

ชื่อรหัสประจำตัว.....ภาควิชา/ชั้นปี.....

1. พิจารณา คลาส Node, LinkNode และ Queue ซึ่งถูกเขียนไว้แล้ว ทำหน้าที่จัดการโหนดด้วยโครงสร้างแบบ Queue (แสดงข้อมูลให้เห็นเท่าที่จำเป็น)

```
class Node { int info ;
            int height , balance ;
            Node left, right;
}
class LinkNode { Node addr;
                LinkNode next;
            public LinkNode (Node ptr) { ..... }
}
class Queue { LinkNode firstq, lastq ;
            public Queue () { ..... }
            void addq(Node bNode) { ..... }
            Node deleteq() { ..... }
}
```

และมีโครงสร้างแบบ binary tree ถูกเขียนขึ้นดังนี้

```
class binaryTree { Node root ;
                public binaryTree () { ..... }
                Node breadthFirstSearch(Node root, int key) {.....}
}
```

ถ้าต้องการค้นหาข้อมูลในต้นไม้ต้นนี้ แบบ Breadth first search โดยต้องการให้ return โหนดที่ค้นเจอ หรือ return null เมื่อค้นไม่เจอ จงเติมคำสั่งให้ method breadthFirstSearch ทำงานได้ตามที่กำหนด

Node breadthFirstSearch (Node root, int key)

```
{ Node current = root;
```

```
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
..... }
```


3) ในการเขียนโปรแกรมเรียงลำดับข้อมูลในอาร์เรย์ แบบ Heap มีการสร้าง class ดังนี้

```
class DataArray { int [ ] data;
                int count;
                .....
public void swap(int i, int j) { ..... }
public void siftDown (int i, int n) { ..... }
public void heapSort ()
{ int i ;
  for (i = (count/2)-1; i >=0; i--)
    siftDown(i, count-1);
  for (i = count-1; i > 0; i--)
  { swap(i, 0); /* Sort data[i] */
    siftDown(0, i-1); }
  }
}
```

data เป็นตัวแปรอาร์เรย์ที่ใช้เก็บข้อมูล เริ่มเก็บตั้งแต่ตัวที่ 0 จนถึง count-1

ในส่วนของเมธอด heapSort ที่กำหนดให้ เป็นปรับปรุงการเรียงลำดับ ให้สามารถเรียงลำดับข้อมูลใน data[] เริ่มตั้งแต่ตัวที่ 0 จนถึง count-1 (จำนวน count ตัว)

ต้องการสร้าง method siftDown () เพื่อปรับปรุงข้อมูลในอาร์เรย์ ตั้งแต่ตัวที่ 0 จนถึงตัวที่ n ให้เป็นแบบ Heap โดยให้ทำงานสอดคล้องกับเมธอด heapSort ที่เขียนขึ้น

คำแนะนำ

1. โจทย์ข้อนี้เป็นการเรียงลำดับข้อมูลตั้งแต่ตัวที่ data[0] ถึง data[count-1]
2. Heap Sort โดยทั่วไปจะเก็บข้อมูลอยู่ในลำดับที่ data[1] ถึง data[count] ดังนั้นจะต้องมีการขจัดเซช โดยการลดหรือเพิ่มหมายเลขตำแหน่งของโหนดที่เกี่ยวข้องกับการเปรียบเทียบให้ถูกต้อง
3. เมธอด siftDown ที่ต้องการให้สร้าง จะต้องขจัดเซชตำแหน่งของโหนดให้ถูกต้อง และสอดคล้องกับการทำงานของเมธอด heapSort

6. จงแสดงรูปต้นไม้ของ Huffman ที่เกิดจากการเข้ารหัสของข้อมูลต่อไปนี้

นามีรุง□นามีรูป□มานีพาดามานา□มานีพาดาดูปู

จากต้นไม้ที่ได้ จงกำหนดรหัสที่ใช้ย่อขนาดของแต่ละตัวอักษร

□	ง	ด	ค	น	ป	พ	ม	ร	า	ี	ู

จำนวนบิตทั้งหมดที่ใช้ในการย่อข้อความที่กำหนดให้ บิต