FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110211 Introduction to Data Structures

YEAR II, First Semester, Mid-term Examination, August 5, 1999, Time 8:30-11:30

หมายเหตุ

1. ไม่อนุญาตให้นำตำราต่างๆ ใดๆ เข้าห้องสอบ
2. ข้อสอบมีทั้งหมด 15 ข้อ 5 หน้า
3. ให้เขียนคำตอบลงในช่องว่างที่เว้นไว้ในกระดาษคำถามนี้ เท่านั้น

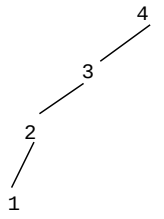
ชื่อ-สกุล.....เลขประจำตัว ..... ตอนเรียนที่ ..... CR58.....

1. อธิบายเหตุผลสั้นๆว่า ทำไมจึงต้องมี iterator class สำหรับการทำงานกับ list ? (2 คะแนน)

2. Java code ข้างล่างนี้คือ class `ListNode` ของ singly linked list จงปรับแก้ class นี้เพื่อให้ใช้กับ doubly linked list ได้ (เขียนทับลงใน code ข้างล่างนี้เลยว่าต้องเพิ่ม หรือลบอะไรบ้าง) (2 คะแนน)

```
class ListNode
{
  ListNode( Object theElement, ListNode n )
  {
    element = theElement;
    next  = n;
  }
  Object  element;
  ListNode  next;
}
```

3. จงเขียนลำดับการ visit nodes ต่างๆ ของ binary search tree ข้างล่างนี้ ด้วยวิธีการ traverse แบบ preorder inorder และ postorder (2 คะแนน)



Preorder : _____.

Inorder : _____.

Postorder : _____.

4. กำหนดให้ binary search tree ต้นหนึ่ง มี 8 nodes
เมื่อนำข้อมูลในต้นไม้นี้มาเรียงลำดับจากน้อยไปมากจะได้ 5, 9, 12, 15, 20, 30, 35, 40
เมื่อ traverse แบบ postorder จะได้ลำดับของ nodes ที่ถูก visited คือ 5, 12, 15, 9, 35, 30, 40, 20
จงวาดต้นไม้ต้นนี้ (ถ้ามีมากกว่าหนึ่งต้นที่ตรงตามข้อกำหนด ให้วาดมาสองต้นก็พอ) (3 คะแนน)

5. นายทักษิณต้องการเขียนโปรแกรม (ชื่อว่า TAIL) เพื่อแสดง k บรรทัดสุดท้ายของแฟ้มข้อมูลที่กำหนดให้ออกสู่จอภาพ ตัวอย่างเช่น TAIL 10 INPUT.TXT หมายถึงการแสดง 10 บรรทัดสุดท้ายของแฟ้ม INPUT.TXT ออกสู่จอภาพ (k มีค่าน้อยมากเมื่อเทียบกับจำนวนบรรทัดที่มีอยู่ในแฟ้ม) จงแนะนำนายทักษิณด้วยว่าควรใช้โครงสร้างข้อมูลอะไรขนาดเล็กๆที่เหมาะกับการทำงานของโปรแกรมนี้ (บรรยายแนวคิดสั้นๆ ไม่ต้องเขียนโปรแกรม) (4 คะแนน)

ใช้ตอบข้อที่ 6 - 7

ใช้ class ต่างๆต่อไปนี้ในการเขียน method เพิ่มเติม 3 methods นิสิตสามารถเรียกใช้ method ที่มีอยู่แล้วใน class ต่างๆข้างล่างนี้ได้ เพื่อเขียน method ใหม่ทั้งสาม

```
Public class ListNode
{
  ListNode( Object theElement )
  {
    this( theElement, null );
  }
  ListNode( Object theElement, ListNode n )
  {
    element = theElement;
    next    = n;
  }
  Object   element;
  ListNode next;
}
```

```
public class LinkedListItr
{
  LinkedListItr( ListNode theNode )
  {
    current = theNode;
  }
  public boolean isPastEnd( )
  {
    return current == null;
  }
  public Object retrieve( );
  public void advance( );

  ListNode current;
}
```

```
public class LinkedList
{
  public LinkedList( )
    { header = new ListNode( null ); }
  public boolean       isEmpty( );
  public void          makeEmpty( );
  public LinkedListItr zeroth( );
  public LinkedListItr first( );
  public void          insert( Object x, LinkedListItr p );
  public LinkedListItr find( Object x );
  public LinkedListItr findPrevious( Object x );
  public void          remove( Object x );
  public static void   printList( LinkedList theList );

  public int           count( );
  public ListNode      remove( LinkedListItr itr );
  public boolean       contentEquivalence( LinkedList P2 );

  private ListNode header;
  ...
}
```

ที่ต้องเขียนเพิ่ม

6. จงเขียน method count เพื่อนับจำนวน node ของ LinkedList (3 คะแนน)

```
//
// usage : int c = list1.count( );
//
public int count( )
{
```

7. จงเขียน method `remove` เพื่อดึง node ที่ระบุด้วย iterator ที่กำหนดให้ ออกจาก linked list `this` (นั่นคือ ลบ node นั้นออกจาก list และ return node ที่ถูกลบ) (3 คะแนน)

```
//
// usage : ListNode node1 = list1.remove( itr1 );
//
public ListNode remove( LinkedListItr itr )
{
```

8. List สอง lists มีคุณสมบัติ content equivalence (CV) เมื่อมีจำนวน node เท่ากัน และ nodes ต่างๆของทั้งสอง lists มีค่าเท่ากันทุก ๆ node โดยที่ตำแหน่งของ node ที่เก็บข้อมูลที่เหมือนกันของทั้งสอง lists อาจแตกต่างกันก็ได้ ตัวอย่างเช่น (1, 3, 5, 6) และ (5, 1, 3, 6) มีคุณสมบัติ CV ในขณะที่ (1, 3, 5, 6) และ (5, 1, 3, 7) ไม่มีคุณสมบัติ CV
จงเขียน method `contentEquivalence` เพื่อตรวจสอบว่า linked list ที่รับเป็น argument กับ linked list `this` มีคุณสมบัติ CV หรือไม่ โดยที่เวลาการทำงานของ method นี้ต้องไม่เกิน $O(nm)$
n และ m คือจำนวนข้อมูลใน list สอง lists ที่ตรวจสอบ (3 คะแนน)

```
//
// usage : if ( list1.contenEquivalence( list2 ) ) ...
//
public boolean contentEquivalence( LinkedList P2 )
{
```

9. class `QueueLi` เป็น queue ชนิดหนึ่งที่สร้าง ด้วย Linked list

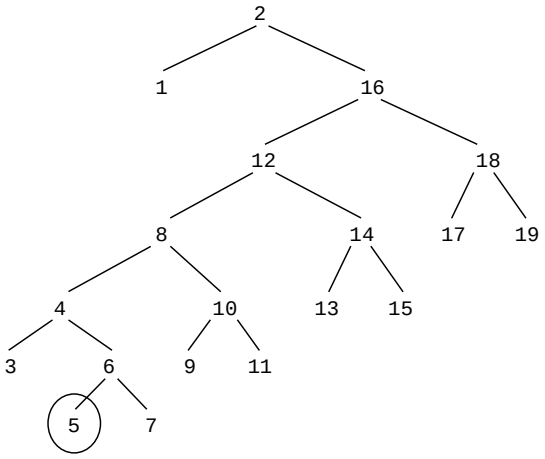
   ก) อธิบายสั้นๆถึงวิธีการสร้าง queue ด้วย linked list ยกตัวอย่างโดยการวาดรูป linked list ที่นำมาใช้สร้าง queue ที่มีข้อมูล (1, 3, 5, 7) โดยที่ 1 คือข้อมูลที่หัวคิว และ 7 คือข้อมูลที่ท้ายคิว (2 คะแนน)

   ข) เขียนชื่อและประเภทของข้อมูลและ methods ต่างๆ ที่จำเป็น (เขียนเฉพาะหัว method ก็พอ ไม่ต้องเขียนรายละเอียดของ method) ที่เป็นสมาชิกของ class `QueueLi` (2 คะแนน)

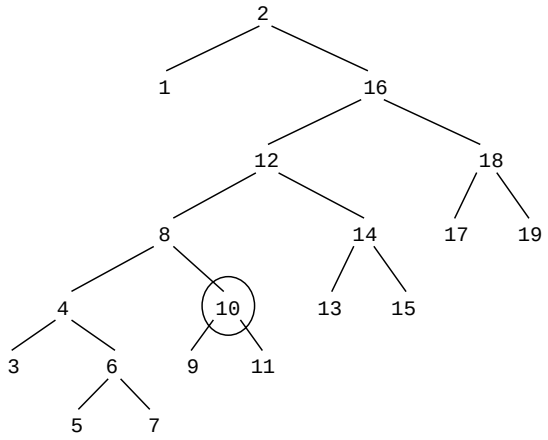
```
public class QueueLi
{
```

   ค) เขียนรายละเอียดของ method `enqueue` ที่มีใน class `QueueLi` (2 คะแนน)

10. ก) จาก splay tree ทางซ้าย จงวาดรูปต้นไม้หลังการทำ splaying ที่ node 5 (2 คะแนน)



ข) จาก splay tree ทางซ้าย จงวาดรูปต้นไม้หลังการทำ splaying ที่ node 10 (2 คะแนน)



ใช้ตอบข้อที่ 11 - 13

กำหนดให้มีการเพิ่มข้อมูล `size` เข้าไปใน `class AvlNode` โดยที่ `size` เป็นสมาชิกของ `AvlNode` ที่เก็บจำนวน nodes ของต้นไม้ย่อยที่ `this` เป็นราก ดังนี้

```
class AvlNode
{
  Comparable  element;
  int         size;
  AvlNode     left;
  AvlNode     right;
  . . .
}
```

11. เมื่อมีสมาชิก `size` ใหม่เพิ่มใน `class AvlNode` ทำให้เราต้องปรับปรุง method ต่างๆที่มีอยู่ใน class `AvlTree` ให้ดูแลข้อมูล `size` นี้ให้ถูกต้องอยู่เสมอด้วย จงปรับปรุง `rotateWithLeftChild` ที่มีอยู่ใน class `AvlTree` (2 คะแนน)

```
public class AvlTree
{
  . . .

  private static AvlNode rotateWithLeftChild( AvlNode k2 )
  {
    AvlNode    k1 = k2.left;

    k2.left = k1.right;
    k1.right = k2;
    k2.height = max( height( k2.left ) + height( k2.right )) + 1;
    k1.height = max( height( k1.left ) + k2.height ) + 1;









  }
}
```

เขียนเพิ่มตรงนี้

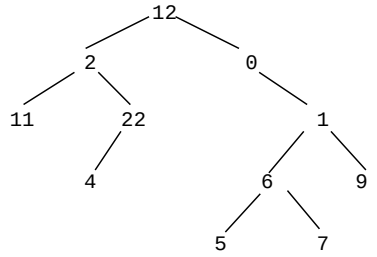
12. จงเขียน method `findKth` ให้กับ `class AvlTree` เพื่อคืนข้อมูลที่มีค่าเล็กที่สุดอันดับที่ k ในต้นไม้ (ถ้า k=1 หมายความว่าต้องการค่าที่เล็กที่สุด) โดยที่ `findKth` รับ parameter 2 ตัว ตัวแรก (k) ระบุอันดับที่เล็กที่สุดที่ต้องการ และพารามิเตอร์ตัวที่สอง (t) ระบุรากของต้นไม้ที่ต้องการค้นหา (5 คะแนน)

```
public class AvlTree
{
  public Comparable findKth( int k )
    { return elementAt( findKth( k, root ) ); }

  private AvlNode findKth( int k, AvlNode t )
  {


  }
}
```

13. ในระหว่างการพัฒนา class เมื่อมีการเพิ่ม method มากขึ้นๆ เรื่อยๆ บางครั้งเราอาจไม่มั่นใจว่า method ต่างๆที่ได้พัฒนาขึ้นนั้น อาจมีบาง method ที่ปรับปรุงข้อมูล `size` ใน `AvlNode` ได้ไม่ถูกต้อง ขอให้นิสิตเขียน method `isSizeValid` เล็กๆ อันหนึ่งที่ตรวจสอบว่าข้อมูล `size` ที่เก็บตาม node ต่างๆในต้นไม้ที่มี `AvlNode t` เป็นราก นั้นมีค่าถูกต้องหรือไม่ (4 คะแนน)

```
  private static boolean isSizeValid( AvlNode t )
  {
```

14. งานๆหนึ่งต้องการพิมพ์ข้อมูลใน nodes ต่างๆของ binary tree จากรากลงมาทีละระดับ ตัวอย่างเช่นต้นไม้ข้างล่างนี้ เมื่อพิมพ์ข้อมูลตาม nodes ต่างๆในแบบที่ต้องการจะได้เป็น 12, 2, 0, 11, 22, 1, 4, 6, 9, 5, 7 จงบรรยายวิธีการทำงานของ method `printTree( BinaryNode t )` ให้แนวคิดของการทำงานที่สั้นๆได้ใจความพร้อมทั้งเขียน Java code ของ `printTree` ประกอบด้วย (ข้อแนะนำ ใช้โครงสร้างข้อมูลที่เรียนมาเพื่อจำ node ที่จะถูกพิมพ์) (3 คะแนน)



15. เราสามารถสร้าง queue ด้วยการใช้ stack 2 ตัว จงเขียน method enqueue และ dequeue ข้างล่างนี้ให้สมบูรณ์ (ที่ทำงาน"เร็วๆ") (4 คะแนน)

```
public class Queue {
  Stack   S1, S2;

  public Queue( )
    { S1 = new Stack( ); S2 = new Stack( ); }
```

`Public enqueue( Object x )`
`{`
 	
`}`
	`public Object dequeue( )`
	`{`
	
	`}`