2FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110211 Introductions to Data Structure

YEAR II, Second Semester, Midterm Examination, December 21, 2010, Time 13:00 – 16:00

ชื่อ-นามสกุล_____เลขประจำตัว | | | | | | | | | 2 | 1 | CR58_____

หมายเหตุ

1. ข้อสอบมีทั้งหมด 11 ข้อในกระดาษคำถามคำตอบจำนวน 8 แผ่น 8 หน้า คะแนนเต็ม 85 คะแนน
2. ไม่อนุญาตให้นำตำราและเครื่องคำนวณต่างๆ ใดๆ เข้าห้องสอบ
3. ควรเขียนตอบด้วยลายมือที่อ่านง่ายและชัดเจน สามารถใช้ดินสอเขียนคำตอบได้
4. ห้ามการหยิบยืมสิ่งใดๆ ทั้งสิ้น จากผู้สอบอื่นๆ เว้นแต่ผู้คุมสอบจะหยิบยืมให้
5. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ ข้อสอบเป็นทรัพย์สินของราชการซึ่งผู้ลักพาอาจมีโทษทางคดีอาญา
6. ผู้ที่ประสงค์จะออกจากห้องสอบก่อนหมดเวลาสอบ แต่ต้องไม่น้อยกว่า 45 นาที
7. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
8. ผู้ที่ปฏิบัติเข้าข่ายทุจริตในการสอบ ตามประกาศคณะวิศวกรรมศาสตร์
   มีโทษ คือ ได้รับ สัญลักษณ์ F ในรายวิชาที่ทุจริต และพักการศึกษาอย่างน้อย 1 ภาคการศึกษา

รับทราบ

ลงชื่อนิสิต (……..……)

หมายเหตุ (เพิ่มเติม)

1. สำหรับข้อที่ให้ออกแบบ หรือ เขียนโปรแกรม คะแนนที่ได้จะแปรตามประสิทธิภาพในการทำงานของโปรแกรม
2. สำหรับข้อที่ให้วิเคราะห์เวลาการทำงาน คะแนนที่ได้จะแปรตามความใกล้เคียงความเป็นจริงของการวิเคราะห์
3. **นิสิตสามารถอ้างถึงและเรียกใช้คลาสต่าง ๆ ที่อยู่ในเอกสารประกอบการสอนได้โดยไม่จำเป็นต้องเขียนขึ้นมาใหม่**
4. **ในข้อที่ต้องออกแบบโครงสร้างข้อมูล นิสิตไม่จำเป็นต้องตรวจสอบถึงกรณีที่มีการใส่ข้อมูลเข้าไปมากกว่าเนื้อที่ที่มีอยู่ (เสมือนว่าการจองพื้นที่นั้นจองมากพอเสมอ) หรือ กรณีที่เอาข้อมูลออกเมื่อไม่มีข้อมูลอยู่ในโครงสร้างข้อมูล**
5. **ให้เขียนคำตอบลงในเฉพาะพื้นที่ที่เว้นว่างไว้**
6. ให้นิสิตเขียนรหัสประจำตัวและเลขที่ใน CR58 ในทุกหน้าของกระดาษคำถามด้วย

---

1. (5 คะแนน) กำหนดให้มีตัวแปร list ซึ่งเป็นอ็อบเจกต์ประเภท ArrayList ขนาด 10 ช่องซึ่งเริ่มต้นไม่มีข้อมูลใดอยู่ จงระบุผลลัพธ์ที่เกิดขึ้นบนหน้าจอ เมื่อได้มีการทำงานตามส่วนของโปรแกรมต่อไปนี้เสร็จเรียบร้อย

```
list.add("A"); list.add("B"); list.add(2,"C"); list.add(1,"D");
list.add("E"); list.remove(2);
list.set(1,list.get(3));
for (int i = 0;i < list.size(); i++) {
  System.out.print(list.get(i)+“ ”);
}
```

……
……
……
……
……
……

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

2. (5 คะแนน) สมมติว่าเราต้องการใส่ข้อมูลเข้าไปใน BinaryHeap (Max Heap) เริ่มต้นที่ยังไม่มีข้อมูล โดยข้อมูลที่ต้องการเพิ่มเข้าไปได้แก่ตัวเลข 1, 2, 3, 4, 5, 6 และ 7 จงบอกลำดับการใส่ข้อมูลที่ทำให้ทุกครั้งที่เรียกการ enqueue นั้น ข้อมูลที่ใส่เข้าไปจะถูก fixUp ไปยังตำแหน่ง root ของ BinaryHeap เสมอ และให้วาดต้นไม้ binary heap หลังการเพิ่มข้อมูลทุกตัวตามลำดับที่ตอบ

.....
.....
.....

3. (5 คะแนน) ปัญหาคูณสามหารสอง ซึ่งมีวิธีการแก้ปัญหาดังเมท็อด bfsM3D2 ด้านล่างนี้ ถ้าเราทำการเรียกเมท็อด bfsM3D2(18) จงระบุค่าที่อยู่ในตัวแปร front, size และ elementData ของอ็อบเจกต์ q เมื่อโปรแกรมทำงานถึงบรรทัดสุดท้าย

```
class Node {
  int value; Node prev;
  Node(int v, Node p) { this.value = v; this.prev = p;  }
  public boolean equals(Object o) { if (!(o instanceof Node)) return false;
    return this.value == ((Node) o).value; }
}
public static void bfsM3D2(int target) {
  Set set = new ArraySet(100);
  Queue q = new ArrayQueue(100);
  Node v = new Node(1,null);
  q.enqueue(v); set.add(v);
  while( !q.isEmpty() ) {
    v = (Node) q.dequeue();
    if (v.value == target) break;
    Node v1 = new Node(v.value/2, v);
    Node v2 = new Node(v.value*3, v);
    if (!set.contains(v1)) {q.enqueue(v1); set.add(v1);}
    if (!set.contains(v2)) {q.enqueue(v2); set.add(v2);}
  }
  if (v.value == target) showSolution(v);
}
```

.....
.....
.....

4. (5 คะแนน) จงระบุว่าข้อย่อยต่อไปนี้ เป็นจริงหรือเป็นเท็จ

- $F(n) = \log(10)\ /\log(n) \in O(\log(10))$ .....
- $F(n) = \ 14n \in O(n^2)$ .....
- $F(n) = \ n(\sqrt{n}) \in O(n \log n)$ .....
- $F(n) = \ (n^3 + 4n^2 - n)\sqrt{n} \in \Theta(n^{3.5})$ .....
- $F(n) = \log(n!) - \log(n) \in \Theta(n \log n)$ .....

5. (5 คะแนน) จงเขียนส่วนของโปรแกรมเพื่อทำการขยายขนาด elementData ในคลาส ArrayQueue แบบวงวนเมื่อขนาดของ elementData ไม่เพียงพอที่จะเก็บข้อมูล

```
public class ArrayQueue implements Queue {
  private Object[] elementData;
  private int size;   private int front;
  // ไม่ได้แสดง method อื่น ๆ ของ ArrayQueue แต่สามารถเรียกใช้ได้ตามปรกติ
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

```
  public void enqueue(Object e) {









    int b = (front + size) % elementData.length;
    elementData[b] = e; size++;
  }
}
```

6. (10 คะแนน) กำหนดให้ ArrayQueue 2 ตัวนั้น “เท่ากัน” ก็ต่อเมื่อแถวคอยทั้ง 2 มีจำนวนข้อมูลเท่ากัน และ การ dequeue ข้อมูลทั้งหมดออกมาจากแถวคอยแรกได้ข้อมูลเหมือนกับการ dequeue ข้อมูลทั้งหมดออกมาจากแถวคอยที่สอง จงปรับปรุงคลาส ArrayQueue โดยให้เพิ่มเมท็อด boolean equals(Object that) โดยที่ that จะเป็นอ็อบเจกต์ประเภท ArrayQueue ซึ่งเมท็อดดังกล่าวจะทำการตรวจสอบว่าใน that นั้น “เท่ากับ” ArrayQueue ที่ทำการเรียกเมท็อดดังกล่าวหรือไม่ นอกจากนี้ หลังจากเรียกเมท็อด equals นั้น ข้อมูลใน that (และในอ็อบเจกต์ที่เรียก) จะต้อง “เท่ากับ” that (และอ็อบเจกต์ที่เรียก) ก่อนเรียก ตัวอย่างเช่น ถ้ามีการเรียก a.equals(b) ติดกันสองครั้ง ผลลัพธ์ที่ได้จะต้องเหมือนเดิม (หมายเหตุ: อ็อบเจกต์ทั้งหลายใน ArrayQueue นี้อาจไม่สามารถเปรียบเทียบว่าใครน้อยกว่า มากกว่าใครได้ จึงไม่สามารถนำไปเรียงลำดับข้อมูล จึงใช้ได้เฉพาะการเปรียบเทียบสองอ็อบเจกต์ว่า เท่าหรือไม่เท่า ด้วย equals เท่านั้น)

```
public class ArrayQueue implements Queue {
  private Object[] elementData;
  private int size;

  // ไม่ได้แสดงเมท็อดต่าง ๆ ของ ArrayQueue แต่สามารถเรียกใช้ได้ตามปรกติ

  public boolean equals(Object that) {
    if (!(that instanceof ArrayQueue)) return false;
    ArrayQueue q2 = (ArrayQueue) that; // ให้เรียกใช้ q2 เมื่อต้องการ access ข้อมูลต่าง ๆ ของ that









  }
}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

7. (10 คะแนน) จงเขียนคลาส StackByQueue ซึ่งเป็นโครงสร้างข้อมูลประเภทกองซ้อน (Stack) โดยที่โครงสร้างข้อมูลนี้จะใช้โครงสร้างข้อมูล ArrayQueue เป็นที่เก็บข้อมูลเท่านั้น ห้ามนิสิตทำการเพิ่มเติม field สมาชิก ใด ๆ ลงในคลาสดังกล่าว ให้นิสิตเขียนโปรแกรมเพิ่มเติมในเมท็อด public void push(Object e), เมท็อด public Object peek() และเมท็อด public Object pop() จากโปรแกรมภาษาจาวาด้านล่างนี้ และให้นิสิตวิเคราะห์ประสิทธิภาพในการทำงานของทั้งสามเมท็อดดังกล่าว นิสิตสามารถสร้างตัวแปรเพิ่มเติมในเมท็อดทั้งสามได้ (**หมายเหตุ: ไม่จำเป็นต้องตรวจสอบกรณีที่มีการใส่ข้อมูลเมื่อกองซ้อนนี้เต็ม หรือการเอาข้อมูลออกเมื่อกองซ้อนนี้ไม่มีข้อมูล และให้ระวังว่าเราไม่สามารถเรียกใช้ตัวแปร elementData ใน ArrayQueue ได้**)

```
public class StackByQueue implements Stack {
    ArrayQueue q;
    StackByQueue(int cap) { q = new ArrayQueue(cap); }
    int size() { return q.size(); }
    int isEmpty() { return q.isEmpty(); }
    public void push(Object e) {









    }
    public Object peek() {









    }
    public Object pop() {









    }
}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

8. (10 คะแนน) สำหรับคลาส ArrayQueue แบบวงวนตามเอกสารประกอบการสอนนั้น เราใช้ตัวแปร front เพื่อบอกถึงตำแหน่งของต้นแถวคอย ถ้าเราเขียนคลาส ArrayQueue ขึ้นมาใหม่โดยใช้ตัวแปร back มาระบุตำแหน่งของท้ายแถวคอย (ตำแหน่งที่จะใส่ข้อมูลเมื่อมีการ enqueue) โดยที่ไม่มีการใช้ตัวแปร front และกำหนดให้เมท็อด enqueue นั้นเป็นดังต่อไปนี้ จงเขียนเมท็อดpeek และ dequeue ที่สามารถทำงานได้อย่างถูกต้อง โดยเขียนเติมลงไปในช่องว่างที่กำหนดให้

```
public class ArrayQueue implements Queue {
  private Object[] elementData;
  private int size; private int back;
  // ไม่ได้แสดง method อื่น ๆ ของ ArrayQueue แต่สามารถเรียกใช้ได้ตามปรกติ
  public void enqueue(Object e) {
    elementData[back] = e; size++; // ให้ถือว่าแถวคอยนี้ไม่เคยเต็ม คือไม่มีการ enqueue เกินขนาด array
    back = (back + 1) % elementData.length;
  }
  public Object peek() {
    if (isEmpty()) throw new NoSuchElementException();









  }
  public Object dequeue() {
    Object e = peek();









    return e;
  }
}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

9. (10 คะแนน) ในโปรแกรม Word Processor ดังเช่น Microsoft Word นั้น จะมีการเก็บรายการประวัติของแฟ้มที่เคยถูกเปิดด้วยโปรแกรมดังกล่าวเพื่อให้ผู้ใช้สามารถเรียกเปิดแฟ้มที่เคยเปิดมาแล้วได้อย่างรวดเร็ว จำนวนแฟ้มในรายการดังกล่าวนั้นจะถูกจำกัดให้มีจำนวนไม่เกินค่าค่าหนึ่ง (กำหนดให้เท่ากับ K) เพื่อประหยัดเนื้อที่ในการแสดงผล ซึ่งโปรแกรม Word Processor จะเลือกเก็บเฉพาะรายชื่อแฟ้มต่างกันที่ถูกใช้ล่าสุดจำนวน K แฟ้มเรียงตามเวลาใช้งานจากล่าสุดไปยังเก่าสุด ตัวอย่างเช่น ถ้ากำหนดให้ K เป็น 3 และเราทำการเปิดแฟ้ม A.doc, B.doc, C.doc, D.doc, C.doc และ D.doc ตามลำดับ ในรายการดังกล่าวจะมีชื่อแฟ้มในรายการประวัติดังนี้ D.doc, C.doc และ B.doc จงออกแบบและเขียนคลาส MostRecentlyUsed ซึ่งต้องมีเมท็อดดังต่อไปนี้

- MostRecentlyUsed(int K) เป็น constructor เพื่อรับค่า K
- void use(String filename) เป็นเมท็อดเพื่อบอกว่าผู้ใช้ได้ทำการเปิดแฟ้มชื่อ filename
- String [] getList() เป็นเมท็อดที่จะคืนอาเรย์ที่ประกอบด้วยชื่อไฟล์ที่ถูกใช้ล่าสุดโดยที่ไฟล์ล่าสุดจะอยู่ที่ตำแหน่ง 0

โดยที่เมท็อด use และ getList นั้นควรจะใช้เวลาในการทำงานเป็น O(K)

คำแนะนำ: ให้ใช้ class ArrayList

```
public class MostRecentlyUsed {









}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

10. (10 คะแนน) ในคลาส BinaryHeap ประเภทฮีปมากสุดนั้น เราสามารถถามหาค่ามากที่สุดได้ในเวลา O(1) ถ้าเราต้องการเพิ่มเมท็อด Object peekMin() เพื่อที่จะขอดูค่าน้อยที่สุดในฮีปดังกล่าวโดยที่กำหนดให้ peekMin นั้นใช้เวลาการทำงานเป็น O(1) เช่นเดียวกัน และให้เมท็อดอื่น ๆ นั้นยังสามารถทำงานได้ตามแบบเดิมและมีประสิทธิภาพเท่าเดิม เราจะต้องทำการแก้ไขคลาส BinaryHeap อย่างไร ให้ระบุแนวคิดพร้อมทั้ง เขียนคลาส BinaryHeap ทั้งคลาส (ถ้าเมท็อดใดเหมือน BinaryHeap ในเอกสารประกอบการสอน ให้เขียนเฉพาะหัวเมท็อดและให้ comment ไว้ว่า “ไม่มีการเปลี่ยนแปลง”)

```
public class BinaryHeap implements PriorityQueue {









}
```

ชื่อ_____นามสกุล_____หมายเลขประจำตัว_____เลขที่ใน CR58_____

11. (10 คะแนน) ปัญหา Range Minimum Query เป็นดังนี้ กำหนดให้มีช่องจำนวน N^2 ช่อง โดยมีหมายเลขกำกับเรียกว่าช่องที่ 0 ถึงช่องที่ $N^2 - 1$ ช่องเหล่านี้สามารถเก็บค่าจำนวนเต็ม (int) ได้ เราสามารถเปลี่ยนแปลงค่าในช่องต่าง ๆ ได้ตามต้องการ และเราต้องการที่จะทราบว่า ค่าที่น้อยที่สุดในช่วงของช่องที่ a ถึงช่องที่ b นั้นเป็นเท่าไร ตัวอย่างเช่น กำหนดให้ N มีค่าเป็น 2 และช่องทั้ง 4 ช่องนั้นมีค่าเป็น 3, 1, -2, 2 ค่าน้อยสุดในช่วงของช่องหมายเลข 1 ถึงช่องหมายเลข 3 นั้นจะมีเค่าเท่ากับ -2
จงออกแบบโครงสร้างข้อมูล RMQ สำหรับแก้ปัญหา Range Minimum Query โดยโครงสร้างข้อมูลนั้นจะต้องมีเมท็อดดังต่อไปนี้

- public RMQ(int N) เป็น constructor เพื่อรับค่า N
- public void set(int idx,int value) ใช้เพื่อบอกว่าค่าของช่องหมายเลข idx นั้นถูกเปลี่ยนเป็นค่า value
- public int getMinimum(int a,int b) ใช้เพื่อถามว่าค่าน้อยสุดในช่วงของช่องหมายเลข a ถึง b นั้นเป็นเท่าไร

การที่จะได้คะแนนมากกว่า 5 คะแนนในข้อนี้ เมท็อด getMinimum จะต้องใช้เวลาไม่เกิน $O(N)$ (ถ้าใช้เวลา $O(N^2)$ จะได้คะแนนไม่เกิน 5 คะแนน)
คำแนะนำ: เราสามารถหาค่าน้อยสุดของตัวแปรจำนวน N ตัวได้ภายในเวลา O(N) ดังนั้น เราควรจะสร้างตัวแปรมา N ตัวสำหรับเก็บค่า minimum ในช่วง 0 ถึง N-1, N ถึง 2N-1, 2N ถึง 3N – 1, …, N*(N-1) ถึง N * N - 1
สำหรับข้อนี้ ถ้าเนื้อที่ไม่พอให้เขียนต่อด้านหลังของหน้านี้ได้