FACULTY OF ENGINEERING
CHULALONGKORN UNIVERSITY
2110211 INTRO. TO DATA STRUCTURES

YEAR II, Second Semester, Final Examination February 22, 2001, Time 9:00–12:00

ชื่อ–สกุล_____ ตอนที่เรียน _____เลขที่ใน CR58_____

หมายเหตุ 1) ไม่อนุญาตให้นำตำราต่างๆ ใดๆ เข้าห้องสอบ
2) ให้เขียนตอบในกระดาษคำถามนี้เท่านั้น
3) ใช้ภาษา Java สำหรับข้อที่ให้เขียนโปรแกรม
4) ข้อสอบมีทั้งหมด 14 ข้อ 5 หน้า

---

1. กำหนดให้ฟังก์ชันแฮชเป็น $h(x) = x \bmod 11$ จงเติมตัวเลขต่อไปนี้ (เริ่มจากตัวซ้ายสุด) ลงในตารางแฮชขนาด 11 ช่อง
5, 8, 4, 15, 16, 27, 33, 22, 13, 25, 18

ก) ใช้ quadratic probing (3 คะแนน)

0	1	2	3	4	5	6	7	8	9	10

ข) ใช้ double hashing โดยให้ $h_2(x) = 7 - (x \bmod 7)$ (3 คะแนน)

0	1	2	3	4	5	6	7	8	9	10

2. จงเติมตัวเลขที่ได้ในแต่ละรอบระหว่างการเรียงข้อมูลใน array ด้วยวิธี quicksort จากน้อยไปมาก กำหนดให้ใช้การ partition ด้วยวิธีที่ได้นำเสนอในชั้นเรียนมา (แบบ median–of–three) โดยตั้ง CUTOFF ไว้ที่ 3 (นั่นคือจะใช้ quicksort เมื่อข้อมูลมีจำนวนเกิน 3 ตัว ในกรณีมีข้อมูลไม่เกิน 3 ตัวให้ใช้ insertion sort) (5 คะแนน)
หมายเหตุ : ให้วงกลมรอบข้อมูลตัวที่ถูกเลือกเป็น pivot ในแต่ละรอบด้วย
ไม่ต้องแสดงการเปลี่ยนแปลงของข้อมูล ระหว่างการ partition และระหว่างการทำงานของ insertion sort

	0	1	2	3	4	5	6	7	8	9	10	11
pass	9	5	7	11	4	8	10	3	6	2	1	10
1												
2												
3												
4												
5												
6												
7												

3. กำหนดให้ array ข้างล่าง เป็น disjoint sets และให้มีการทำ path compression ใน method find( ) จงเติมตัวเลขลงในตารางแสดงผลลัพธ์ของ find(7) และให้วาดรูป trees ของ disjoint sets ทั้งก่อนและหลัง find(7) ประกอบทางด้านขวาของ array ด้วย (3 คะแนน)

	0	1	2	3	4	5	6	7	8	9
ก่อน	1	–1	0	4	2	6	4	8	6	7

หลัง										

เลขประจำตัว _____

4. จงแสดงข้อมูลใน array ที่แทน binary heap ระหว่างการทำ BuildHeap (ชนิด Min Heap) จากข้อมูลใน array ข้างล่างนี้ วาดต้นไม้ประกอบทางด้านขวาของ array ด้วย (3 คะแนน)

78	61	75	34	23	55	46

5. `find2ndMin()` คืนค่าของข้อมูลตัวที่น้อยที่สุดเป็นอันดับที่สองใน BinaryHeap (แบบ min heap ตามที่ได้เรียนมา) โดยที่ method นี้ไม่ทำให้ heap มีการเปลี่ยนแปลง จงเติม method นี้ซึ่งใช้เวลาการทำงานคงตัวให้สมบูรณ์ (หมายเหตุ : ให้ระวังกรณีพิเศษต่างๆ ที่อาจเกิดขึ้น) (5 คะแนน)

```
public Comparable find2ndMin( )
{
   if ( isEmpty() ) return null;









}
```

6. สมพรเป็นคนพูดน้อย แต่พูดครั้งใดเพื่อน ๆ เป็นต้องขำกลิ้งไปทุกที วิทูรย์จึงคิดทำ "คลังคำคม" ของสมพรขึ้นมาให้สามารถค้นหาคำคมได้ตามหัวเรื่องหรือคำสำคัญโดยใช้ hash table จงแสดง class สำหรับคลังคำคมนี้ (เขียนเฉพาะส่วนที่เป็นข้อมูล ไม่ต้องเขียน method) โดยกำหนดให้ใช้หลักการของ open addressing สำหรับ collision (5 คะแนน)

7. ในการทำ maze โดยใช้หลักการของ disjoint set นั้น algorithm สามารถจะหยุดได้เมื่อทางเข้าและทางออกตกอยู่ใน equivalent set เดียวกัน ถ้าเมื่อ algorithm หยุดลง (ทำสำเร็จ) แต่มี set เหลือมากกว่าหนึ่ง set จะหมายถึงอย่างไร(5 คะแนน)

8. ทำไมลักษณะของ merge sort และ quick sort จึงรวมเรียกว่าเป็นระบบ divide and conquer และ sort แต่ละแบบมีข้อเสีย (หรือข้อควรระวัง) ที่สำคัญอย่างไร (5 คะแนน)

9. จงเติมส่วนของโปรแกรมในวงเล็บของคำสั่ง for ที่เว้นว่างไว้ให้สมบูรณ์ (โปรแกรมนี้คือการเรียงลำดับข้อมูลจากน้อยไปมากด้วยวิธี Shellsort) (5 คะแนน)

```
public static void shellsort( Comparable [ ] a )
{
    int j;

    for( int gap = a.length / 2; gap > 0; gap /= 2 )
        for( int i = gap; i < a.length; i++ )
        {
            Comparable tmp = a[ i ];
            for(                                    )
                a[ j ] = a[ j – gap ];
            a[ j ] = tmp;
        }
}
```

```
for(                                                    )
```

10. จงวาดรูปร่างของต้นไม้ซึ่งสูงที่สุด ที่ได้จากการ union disjoint sets ต่างๆ เป็นจำนวน 15 ครั้ง โดยตอนเริ่มต้นมี 16 เซต แต่ละเซตมีสมาชิกหนึ่งตัว กำหนดให้การ union นั้นใช้วิธี union–by–height (ไม่ต้องแสดงรายละเอียดระหว่างการ union แสดงผลลัพธ์หลังจากการ union ครั้งสุดท้ายก็พอ) (3 คะแนน)

11. คุณเหนาะบอกว่าเราสามารถหาค่ามากที่สุดใน min–heap ได้ง่ายๆ โดยการวิ่งไล่เปรียบเทียบหาตัวมากสุดเฉพาะครึ่งขวาของชุดข้อมูลใน array ก็พอ (ตั้งแต่ตัวที่ $1+\lfloor n/2 \rfloor$ ถึง n โดยที่ n คือจำนวนข้อมูลใน heap) ไม่ต้องไปสนใจชุดข้อมูลครึ่งซ้ายเลย นอกจากนี้คุณเหนาะยังอวดด้วยว่านี่เป็นวิธีที่ดีที่สุดแล้ว จงให้เหตุผลสนับสนุนหรือคัดค้านคำกล่าวอ้างข้างต้น (ถ้าคัดค้านให้นำเสนอแนวทางของวิธีที่ดีกว่าด้วย ไม่ต้องเขียนโปรแกรม) (5 คะแนน)

เลขประจำตัว _____

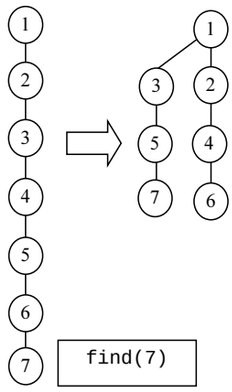
12. คุณหน่อยแนะนำวิธีแก้ไขปัญหาการชนแบบใหม่ดังนี้ กำหนดให้ตารางแฮชมีขนาด m ช่อง $h(x)$ คือฟังก์ชันแฮช และใช้สูตร $h_i(x) = (h(x) + F(i)) \bmod m$ ในการคำนวณช่องที่ต้อง probe ในตารางแฮชหลังการชนครั้งที่ i โดยที่ให้ $F(0) = 0$ และ $F(1), F(2), ..., F(m-1)$ คือ random permutation ของเลข 1 ถึง $m-1$ (ซึ่งเตรียมแบบ random ไว้ตอน construct ตารางแฮช หมายความว่าตารางแฮชสองตารางอาจมีค่า $F(i)$ ต่างกันได้) อยากทราบว่า

ก) วิธีนี้จะ probe พบช่องว่างในตารางแฮชหรือไม่ ถ้าข้อมูลยังไม่เต็มตาราง อธิบายเหตุผลประกอบด้วย (2 คะแนน)

ข) วิธีนี้จะแก้ปัญหา primary clustering หรือไม่ อธิบายเหตุผลประกอบด้วย (2 คะแนน)

ค) วิธีนี้จะแก้ปัญหา secondary clustering หรือไม่ อธิบายเหตุผลประกอบด้วย (2 คะแนน)

13. path compression ช่วยให้ต้นไม้มีความสูงลดลงได้หลังจากการ find แต่วิธีการโปรแกรมเพื่อทำ path compression ที่ได้นำเสนอในชั้นเรียนนั้น ถึงแม้จะสั้นแต่ก็เป็นแบบ recursive ซึ่งมีขั้นตอนการวิ่งไล่จากโหนดที่ถูกหาย้อนไปถึงราก และมีการวิ่งจากรากลงกลับมายังโหนดที่ถูกหานั้น เพื่อเปลี่ยนพ่อใหม่ของโหนด การวิ่งขึ้นลงสองรอบนี้เป็นผลมาจากการเขียนโปรแกรมแบบ recursive ได้มีผู้นำเสนอให้เปลี่ยนพ่อใหม่ด้วยวิธี path halving คือแทนที่จะเปลี่ยนพ่อใหม่ให้เป็นรากของต้นไม้ ก็เพียงแต่ให้พ่อใหม่เป็นโหนดที่เคยเป็นปู่ (สำหรับกรณีที่เดิมมีปู่ ถ้าไม่มีก็ไม่ต้องเปลี่ยนพ่อ) ดังรูป



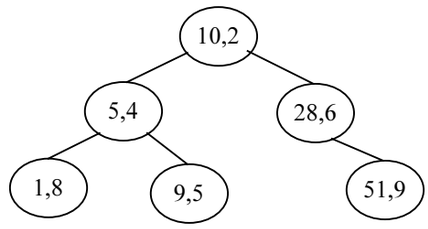
จงเขียน method find (ด้วยภาษา Java อย่างละเอียด) แบบ path halving โดยเขียนแบบวงวนธรรมดา (ใช้ for หรือ while ไม่ใช่แบบ recursive)
กำหนดให้ใช้วิธีการแทน disjoint sets ด้วย array ที่ได้เรียนมา (5 คะแนน)

```
public int find( int x )
{




}
```

14. treap เป็น binary search tree แบบหนึ่งซึ่งในแต่ละ node นอกจากจะเก็บข้อมูล (เพื่อใช้ในการค้นหา ซึ่งเป็นไปตามกฎของ binary search tree) แล้วยังเก็บข้อมูลเสริมอีกตัวหนึ่ง โดยถ้าหากเราพิจารณาที่ข้อมูลเสริมต่างๆ ในต้นไม้ จะเห็นได้ว่าความสัมพันธ์ของข้อมูลเสริมเหล่านี้ถูกจัดเก็บตามลักษณะเดียวกับ min–heap ดังรูป (จำนวนทางซ้ายใน node คือข้อมูลเพื่อการค้นหา และจำนวนทางขวาใน node คือข้อมูลเสริมที่กล่าวถึง)



ก) จงวาดรูปของ treap ซึ่งเก็บข้อมูลดังนี้ (1,10), (2,8), (3,6), (4,9), (5,7), (6,15), (7,4), (8,1), (9,20), (10,2)
(3 คะแนน)

ข) จงบรรยายขั้นตอนการเพิ่มข้อมูลใหม่ใน treap (ไม่ต้องเขียนโปรแกรม) พร้อมทั้งยกตัวอย่างการทำงานเมื่อมีการเพิ่ม (8,0) ใน treap ที่แสดงข้างบนนี้ (5 คะแนน)