รหัสนักศึกษา_____

ให้นักศึกษาตอบคำถามต่อไปนี้

[4 คะแนน] 1. เปรียบเทียบข้อดีข้อเสียของการใช้ Big O กับการ benchmark

2. การคำนวณ Big O

[2 คะแนน] 2.1 หากนักศึกษาต้องการคำนวณ Big O ของขั้นตอนวิธีในการคำนวณปริมาณการใช้น้ำมันของรถยนต์ส่วนตัวทั้งหมดในประเทศไทยในเดือนตุลาคม นักศึกษาจะใช้อะไรเป็นขนาดของปัญหา (n)

[2 คะแนน] 2.2 สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O(\sqrt{n})$ หากขนาดของปัญหาเพิ่มขึ้นเป็น 100 เท่า (จาก n เป็น 100n) แล้ว ขั้นตอนวิธีนี้จะใช้เวลาเพิ่มขึ้นเป็นกี่เท่า

[2 คะแนน] 2.3 สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น O(log n) หากขั้นตอนวิธีนี้ใช้เวลา 100ms เมื่อ n=10 แล้วขั้นตอนวิธีนี้จะใช้เวลาประมาณเท่าไหร่เมื่อ n=1000

รหัสนักศึกษา_____

3. จากโปรแกรมหา $\lceil\sqrt{n}\rceil$ ต่อไปนี้ (จำนวนเต็มที่น้อยที่สุดที่มากกว่ารากที่ 2 ของ n)

```
int CeilRootN1 (int n) {
  int r=1;
  while(r*r<n) {
    r++;
  }
  System.out.println(r);
}
```

```
int CeilRootN2(int n) {
  int a=1, b=n, r;
  while(b-a>0) {
    r = (a+b)/2;
    if( (r*r>=n) || ((r-1)*(r-1) <n) break;
  }
  System.out.println(r);
}
```

[5 คะแนน] 3.1 ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ CeilRootN1 และสรุปว่า CeilRootN1 มี Big O เป็นเท่าใด

[5 คะแนน] 3.2 ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ CeilRootN2 และสรุปว่า CeilRootN2 มี Big O เป็นเท่าใด

รหัสนักศึกษา_____

4. กำหนดให้ int a[] = {8,7,6,5,4,3,2,1,0};

[2 คะแนน] 4.1 หากสั่ง System.out.println(a[a[4]-a[2]]); แล้ว จะได้ผลลัพธ์ออกมาเป็นอะไร

[2 คะแนน] 4.2 หากสั่ง System.out.println(a[a[8]-a[5]]); แล้ว จะได้ผลลัพธ์ออกมาเป็นอะไร

5. กำหนดให้ linked list มีโครงสร้างดังนี้

first → [1] → [2] → [3] →[4] →[5] →[6] →[7] → null

[2 คะแนน] 5.1 หากสั่ง p=first; p=p.next; System.out.println(p.next.data) แล้ว จะได้ผลลัพธ์ออกมาเป็นอะไร

[2 คะแนน] 5.2 หากสั่ง p=first; p.next.next=p.next.next.next.next แล้ว จะได้ผลลัพธ์ linked list โครงสร้างแบบใด

รหัสนักศึกษา_____

6. จากส่วนของโปรแกรมต่อไปนี้

```
int maxIndex = 0;
for(int i=1; i<a.length; i++) {
        if(a[i-1]>a[i]) {
                int tmp = a[i-1];
                a[i-1] = a[i];
                a[i] = tmp;
        }
}
System.out.println(a[a.length-1]);
```

[5 คะแนน] 6.1 อธิบายการทำงานของส่วนของโปรแกรมนี้ พร้อมทั้งแสดงผลของโปรแกรม

[5 คะแนน] 6.2 ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดส่วนของโปรแกรมนี้ พร้อมสรุปว่า Big O เป็นเท่าใด

รหัสนักศึกษา_____

7. จากโปรแกรมย่อยต่อไปนี้ (กำหนด class Node ของ linked list)

```
void methodA(Node p) {
        if( p==null || p.next==null ) return
        Node q = p.next;
        Node m = q;
        while(q!=null) {
                if(q.data<m.data) m=q;
                q = q.next;
        }
        int tmp = p.data;
        p.data = m.data;
        m.data = tmp;
}
```

[4 คะแนน] 7.1 หากกำหนดให้ linked list มีข้อมูลเป็น first → [3] → [5] → [7] → [11] → null แล้ว หลังจากการเรียก methodA(first) แล้ว โครงสร้าง linked list จะกลายเป็นอย่างไร

[4 คะแนน] 7.2 Big O ของโปรแกรมย่อยนี้เป็นเท่าไร

[4 คะแนน] 7.3 หากเรียกใช้ methodA ตามส่วนของโปรแกรมด้านล่างแล้ว ผลของ linked list จะออกมาเป็นอย่างไร

```
node p = first;
while(p!=null) {
        methodA(p);
        p = p.next;
}
```

รหัสนักศึกษา_____

[2 คะแนน] 8.1 การดำเนินการใดบ้างของสแตกที่ต้องใช้เวลาเป็น O(1) เสมอ

[10 คะแนน] 8.2 กำหนดให้มีสแตกว่างเปล่าอยู่ 2 สแตก ชื่อว่า stackA และ stackB ให้นักศึกษาเขียนข้อมูลภายในสแตก (ให้ข้อมูลขวาสุดเป็นด้านบนของสแตก) ที่ผ่านการดำเนินการต่อไปนี้

8.2.1 stackA.push(5) stackA: stackB:	8.2.2 stackB.push(8) stackA: stackB:
8.2.3 stackA.push(9) stackA: stackB:	8.2.4 stackB.push(stackA.pop()) stackA: stackB:
8.2.5 stackA.push(stackB.pop()-stackA.pop()) stackA: stackB:	8.2.6 stackA.push(7) stackA: stackB:
8.2.7 stackB.push(9) stackA: stackB:	8.2.8 stackA.push(stackB.pop()*stackB.pop()) stackA: stackB:
8.2.9 stackA.push(stackA.pop()) stackA: stackB:	8.2.10 stackB.push(stackA.top()+1) stackA: stackB:

รหัสนักศึกษา_____

[2 คะแนน] 9.1 การดำเนินการใดบ้างของ queue ที่ต้องใช้เวลาเป็น O(1) เสมอ

[10 คะแนน] กำหนดให้มี queue ว่างเปล่าอยู่ 2 queue ชื่อว่า qA และ qB ให้นักศึกษาเขียนข้อมูลภายใน queue (ให้ข้อมูลซ้ายสุดเป็นด้านหน้า queue) ที่ผ่านการดำเนินการต่อไปนี้

9.2.1 qA.enqueue(2) qA: qB:	10.2 qB.enqueue(3) qA: qB:
9.2.3 qA.enqueue(4) qA: qB:	9.2.4 qB.enqueue(qA.dequeue()) qA: qB:
9.2.5 qA.enqueue(5) qA: qB:	9.2.6 qB.enqueue(6) qA: qB:
9.2.7 qA.enqueue(qA.dequeue()*qB.dequeue()) qA: qB:	9.2.8 qB.enqueue(qA.dequeue()*qA.dequeue()) qA: qB:
9.2.9 qA.enqueue(qA.dequeue()+2) qA: qB:	9.2.10 qB.enqueue(qB.dequeue()-2) qA: qB:

รหัสนักศึกษา_____

จากนิพจน์ทางคณิตศาสตร์ต่อไปนี้ 9 – 8 * (7 + 6 – 5) - 4 / 3 + 2 ให้วาดข้อมูลใน stack และ queue

[3 คะแนน] 10.1 หลังจากดำเนินการด้วย Shunting Yard Algorithm ผ่านวงเล็บปิดแล้ว

[3 คะแนน] 10.2 หลังจากดำเนินการด้วย Shunting Yard Algorithm ผ่านเลข 3 แล้ว

รหัสนักศึกษา_____

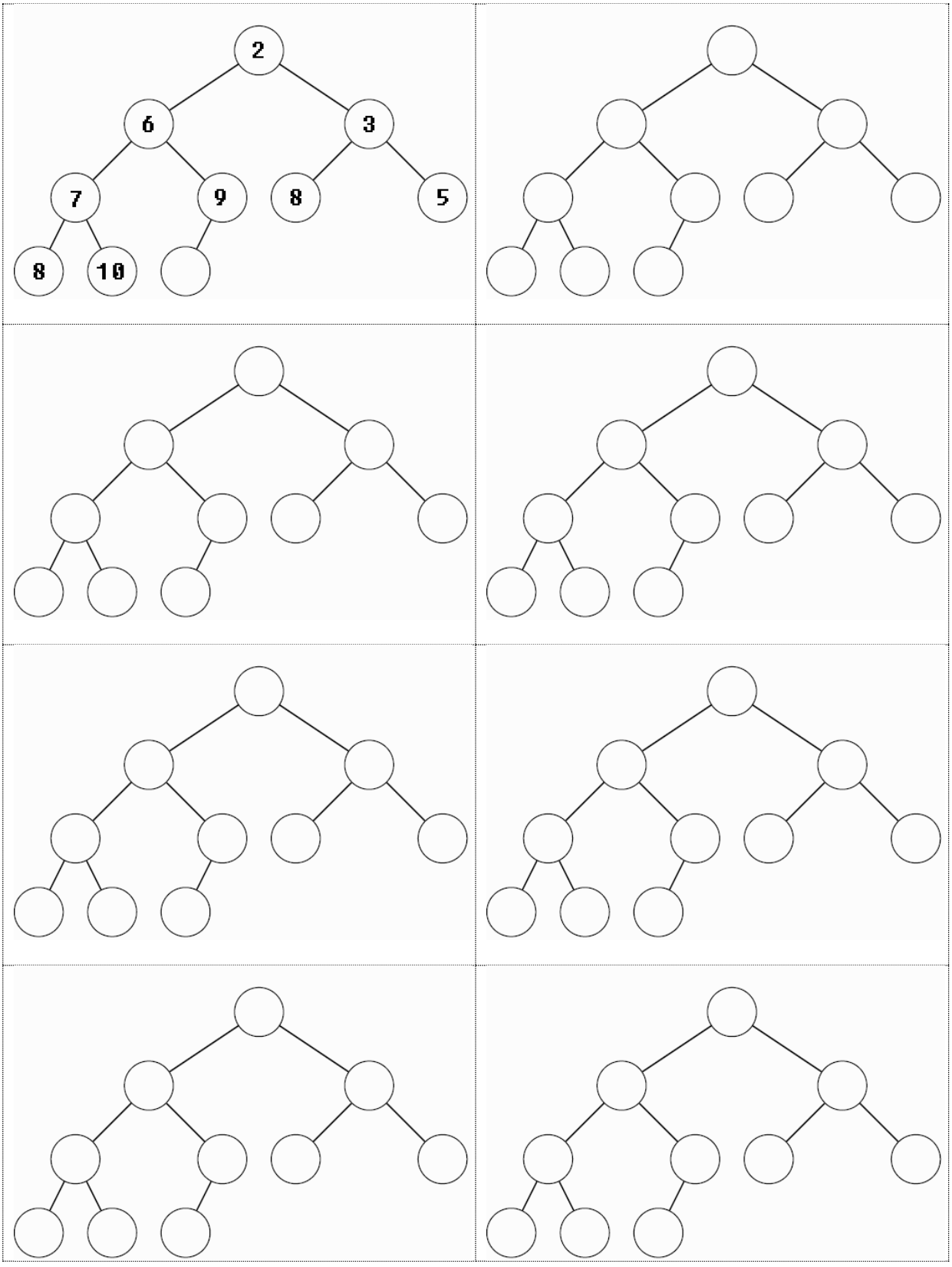
จากนิพจน์ Reversed Polish Notation นี้ 1 2 3 * + 4 - 5 * 6 / ให้วาดข้อมูลใน stack และ queue

[3 คะแนน] 11.1 หลังจาก RPN Evaluation Algorithm ดำเนินการผ่านเลข 4 แล้ว

[3 คะแนน] 11.2 หลังจาก RPN Evaluation Algorithm ดำเนินการผ่านตัวดำเนินการ * ตัวที่สองแล้ว

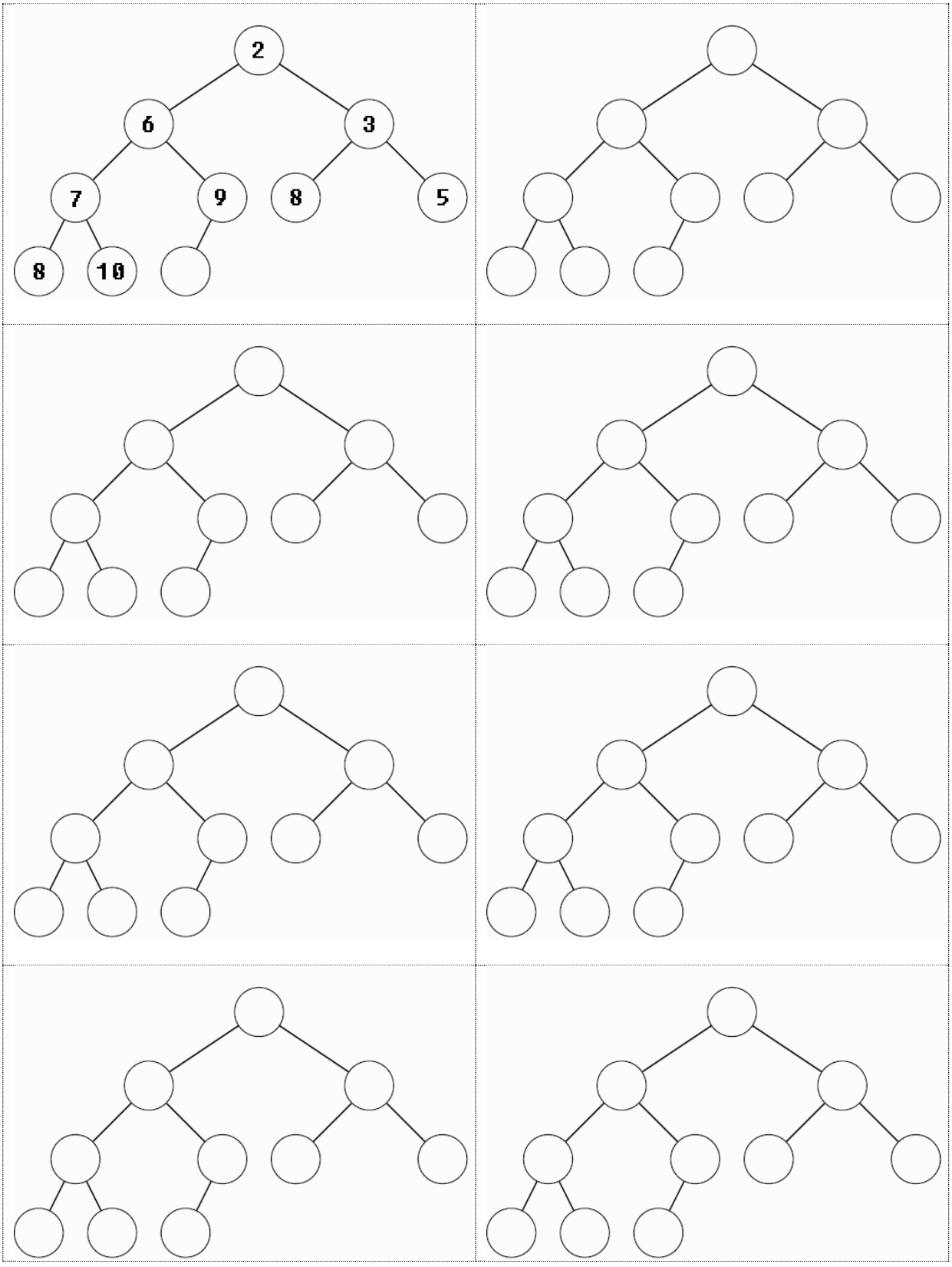
รหัสนักศึกษา_____

[4 คะแนน] 12. จงเขียนลำดับขั้นในการ dequeue จาก heap นี้



รหัสนักศึกษา_____

[4 คะแนน] 13. จงเขียนลำดับขั้นในการ enqueue(4) ลงบน heap นี้



รหัสนักศึกษา_____

กำหนดให้สตริงก์ A มี Hidden Reverse อยู่ใน B เมื่อสตริงก์ B มีตัวอักษรของ A อยู่ในลำดับที่สลับกลับทาง ซึ่งตัวอักษรจาก A นั้น อาจจะแทรกอยู่ในตัวอักษรอื่นๆ ใน B ได้ เช่น

A: **NIMIC** B: **C**AI**B**U**M**C**IN**

ซึ่งจะเห็นได้ว่า ตัวอักษร NIMIC จาก A มีแทรกอยู่ใน B ทั้งหมด แต่สลับจากหลังมาหน้า

[10 คะแนน] 14. ให้เขียนโปรแกรมตรวจสอบ Hidden Reverse และวิเคราะห์ว่ามีค่า Big O เป็นเท่าใด

คำแนะนำ 1 เขียนเป็น pseudo code หรือ flow chart และวิเคราะห์ สามารถได้ 8/10 คะแนน

คำแนะนำ 2 ใช้ stack (ไม่ต้องเขียน implementation ของ stack)

คำแนะนำ 3 คำสั่งดูอักษรตำแหน่งที่ i ใน String s คือ s.charAt(i)