รหัสนักศึกษา_____

1. Big O vs. Benchmark

1.1 [2 คะแนน] ให้ยกตัวอย่างสถานการณ์ที่สามารถวัดประสิทธิภาพได้ด้วย Benchmark และ Big O

1.2 [2 คะแนน] จากข้อ 1.1 หากใช้ Big O เพื่อวัดประสิทธิภาพเพียงอย่างเดียว จะมีข้อดีข้อเสียเพิ่มขึ้นไหร่ไม่อย่างไร

2. การคำนวณ Big O

2.1 [2 คะแนน] หากนักศึกษาต้องการคำนวณ Big O ของขั้นตอนวิธีในการคำนวณหารายได้ทั้งหมดของแท็กซี่ในกรุงเทพฯ นักศึกษาจะทำอย่างไร และใช้อะไรเป็นขนาดของปัญหา (n)

2.2 [2 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O[n^2 \log n]$ หากขนาดของปัญหาเพิ่มขึ้นเป็น 20 เท่า (จาก n เป็น 20n) แล้ว ขั้นตอนวิธีนี้จะใช้เวลาเพิ่มขึ้นเป็นกี่เท่า

2.3 [2 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O(\sqrt[3]{n})$ หากขั้นตอนวิธีนี้ใช้เวลา 10ms เมื่อ n=100 แล้วขั้นตอนวิธีนี้จะใช้เวลาประมาณเท่าไหร่เมื่อ n=1000

รหัสนักศึกษา_____

3. จากโปรแกรมหา $\lfloor\sqrt{n^2}\rfloor$ ต่อไปนี้ (จำนวนเต็มที่มากที่สุดที่น้อยกว่ากำลังสองของ n)

```
int floorSqrNSqr1 (double n) {
  int r = 1;
  while(r<Math.sqrt(n*n)) {
    r++;
  }
  return r-1;
}
```

```
int floorSqrNSqr2(double n) {
  int i = 1
  int r = i*i;
  while(r<n*n) {
    i++;
    r = i*i;
  }
  return r-1;
}
```

3.1 [5 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ floorSqrNSqr1 และสรุปว่า floorSqrNSqr1 มี Big O เป็นเท่าใด

3.2 [5 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ floorSqrNSqr2 และสรุปว่า floorSqrNSqr2 มี Big O เป็นเท่าใด

รหัสนักศึกษา_____

4. [6 คะแนน] ให้เขียนโปรแกรมเพื่อให้ b[i] เป็น ผลบวกของ a[0] ถึง a[i]

ตัวอย่าง หาก a = { 0, 2, 1, 0, 9, 4 } จะได้ b = { 0, 2, 3, 3, 12, 16 }

5. [6 คะแนน] ให้เขียนฟังก์ชัน (method) ชื่อ void cut(int index) เพื่อทำให้ Node ที่ index ของรายการโยง (linked list) กลายเป็นตัวแรกของรายการโยง และให้นำข้อมูลก่อนหน้าทั้งหมด ไปต่อที่ด้านหลัง

ตัวอย่าง จาก head→[0]→[1]→[2]→[3]→[4]→null

เรียก cut(2) เปลี่ยนเป็น head→[2]→[3]→[4]→[0]→[1]→null

รหัสนักศึกษา_____

6. จากส่วนของโปรแกรมต่อไปนี้

```
int index = 0;
int b = new int[a.length];
b[0] = a[0];
for(int i=1; i<a.length; i++) {
    if(a[i]>b[i-1]) b[i] = a[i];
    else b[i] = b[i-1];
}
for(int i=0; i<b.length) i++)
    System.out.print(b[0]+" ");
```

[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 method7(Node p) {
    while(p!=null) {
        if(p.next!=null) {
            p.data = p.data + p.next.data;
            p.next = p.next.next;
        }
        p = p.next;
    }
}
```

7.1 [4 คะแนน] หากกำหนดให้ linked list มีข้อมูลเป็น head → [2] → [1] → [3] → [7] → [4] → null แล้ว หลังจากการเรียก method7(head) โครงสร้าง linked list จะกลายเป็นอย่างไร

7.2 [4 คะแนน] Big O ของโปรแกรมย่อยนี้เป็นเท่าไร

7.3 [4 คะแนน] หากเรียกใช้ method7 ตามส่วนของโปรแกรมด้านล่างแล้ว ผลของ linked list จะออกมาเป็นอย่างไร

```
while(head.next!=null) method7(head);
```

รหัสนักศึกษา_____

8.1 [2 คะแนน] การดำเนินการใดบ้างของสแตกที่ต้องใช้เวลาเป็น O(1) เสมอ

8.2 [5 คะแนน] กำหนดให้มีสแตกว่างเปล่าอยู่ 2 สแตก ชื่อว่า stackA และ stackB ให้นักศึกษาเขียนข้อมูลภายในสแตก (ให้ข้อมูลขวาสุดเป็นด้านบนของสแตก) ที่ผ่านการดำเนินการต่อไปนี้

8.2.1 stackA.push(1) stackA: stackB:	8.2.2 stackB.push(2) stackA: stackB:
8.2.3 stackA.push(3) 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(1) stackA: stackB:
8.2.7 stackB.push(4) 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_____

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

9.2 [5 คะแนน] กำหนดให้มี queue ว่างเปล่าอยู่ 2 queue ชื่อว่า qA และ qB ให้นักศึกษาเขียนข้อมูลภายใน queue (ให้ข้อมูลซ้ายสุดเป็นด้านหน้า queue) ที่ผ่านการดำเนินการต่อไปนี้

9.2.1 qA.enqueue(1) qA: qB:	10.2 qB.enqueue(2) qA: qB:
9.2.3 qA.enqueue(3) qA: qB:	9.2.4 qB.enqueue(qA.dequeue()) qA: qB:
9.2.5 qA.enqueue(4) qA: qB:	9.2.6 qB.enqueue(5) 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()+6) qA: qB:	9.2.10 qB.enqueue(qB.dequeue()-7) qA: qB:

รหัสนักศึกษา_____

10. จากนิพจน์ทางคณิตศาสตร์ต่อไปนี้ 2+ 9 / 3 * ((5 + 2 - 4 * 1) * (6 – 8 / 2)) ให้วาดข้อมูลใน stack และ queue

10.1 [3 คะแนน] หลังจากดำเนินการด้วย Shunting Yard Algorithm ผ่านเลข 5 แล้ว

10.2 [3 คะแนน] หลังจากดำเนินการด้วย Shunting Yard Algorithm ผ่านเลข 8 แล้ว

รหัสนักศึกษา_____

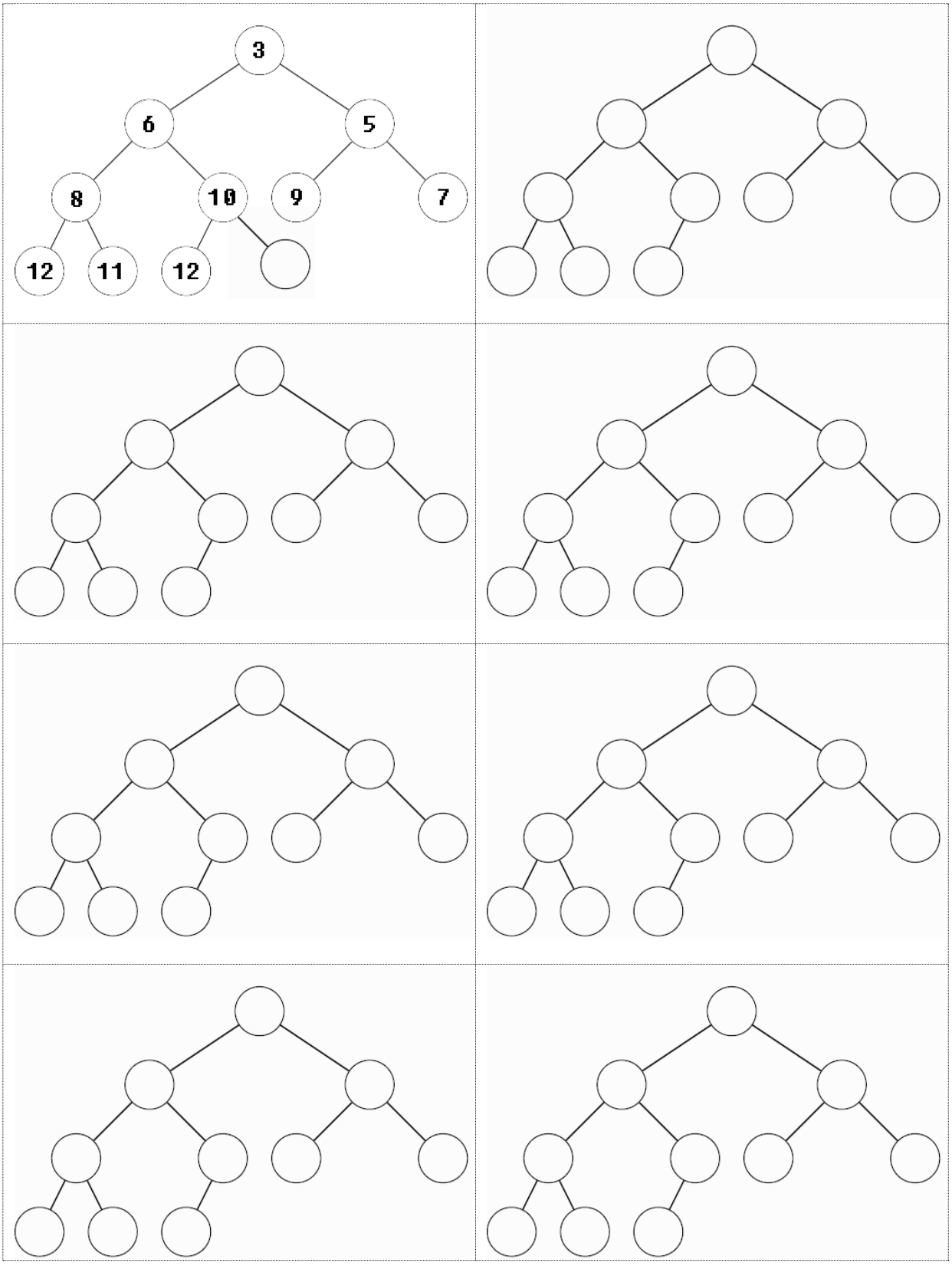
11. จากนิพจน์ Reversed Polish Notation นี้ 3 5 + 9 7 1 * 8 2 / - / - ให้วาดข้อมูลใน stack

11.1 [3 คะแนน] หลังจาก RPN Evaluation Algorithm ดำเนินการผ่านเลข 7 แล้ว

11.2 [3 คะแนน] หลังจาก RPN Evaluation Algorithm ดำเนินการผ่านตัวดำเนินการ - ตัวที่แรกแล้ว

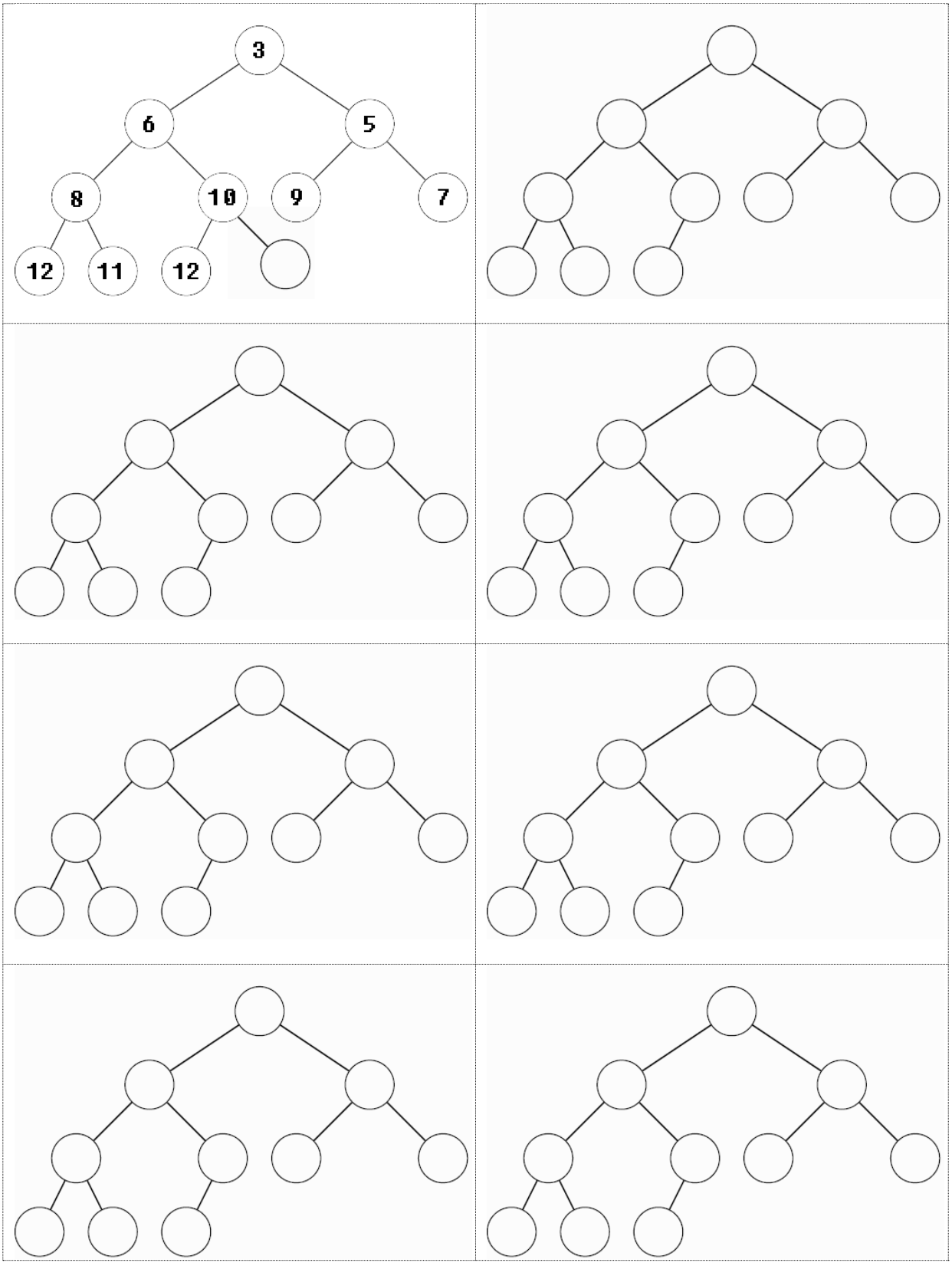
รหัสนักศึกษา_____

12. [4 คะแนน] จงเขียนลำดับขั้นในการ enqueue(4) ลงบน heap นี้



รหัสนักศึกษา_____

13. [4 คะแนน] จงเขียนลำดับขั้นในการ dequeue() จาก heap นี้



รหัสนักศึกษา_____

14. [10 คะแนน] กำหนดให้ A และ B เป็นอาร์เรย์ของตัวเลขที่ไม่ซ้ำและไม่มีการเรียงลำดับ ให้เขียนโปรแกรมเพื่อนับจำนวนสมาชิกที่ซ้ำกันของ A และ B และวิเคราะห์ว่ามีค่า Big O เป็นเท่าใด

คำแนะนำ 1 เขียนเป็น pseudo code หรือ flow chart และวิเคราะห์ สามารถได้ 6/10 คะแนน

คำแนะนำ 2 ใช้ heap (priority queue) (ไม่ต้องเขียน implementation ของ heap/priority queue)

คำแนะนำ 3 คะแนนขึ้นอยู่กับประสิทธิภาพ (Big O) ด้วย