รหัสนักศึกษา_____

1 ให้นักศึกษาตอบคำถามต่อไปนี้

1.1 [2 คะแนน] อะไรเป็นเหตุผลที่ต้องใช้ Big O ในการวิเคราะห์ประสิทธิภาพของโปรแกรม

1.2 [2 คะแนน] หากนักศึกษาต้องการคำนวณ Big O ของขั้นตอนวิธีในการค้นหาคนหนึ่งคนจากประชากรในประเทศใดประเทศหนึ่ง นักศึกษาจะใช้อะไรเป็นขนาดของปัญหา (n)

1.3 [2 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O(n^2)$ หากขนาดของปัญหาเพิ่มขึ้นเป็น 2 เท่า (จาก n เป็น 2n) แล้ว ขั้นตอนวิธีนี้จะใช้เวลาเพิ่มขึ้นเป็นกี่เท่า

1.4 [2 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O(n^3)$ หากขั้นตอนวิธีนี้ใช้เวลา 100ms เมื่อ n=10 แล้ว ขั้นตอนวิธีนี้จะใช้เวลาเป็นเท่าไหร่เมื่อ n=50

รหัสนักศึกษา_____

2 จากโปรแกรมสำหรับพิมพ์จำนวนเต็มที่น้อยกว่า n และสามารถหารากที่ 2 ออกมาเป็นจำนวนเต็มต่อไปนี้

```
class PerfectSquare1 {
   static int n=1000;
   public static void main(String args[]) {
      for(int i=1; i<n; i++) {
         int r = (int)Math.sqrt(i);
         if(r*r==i) {
            System.out.println(r+"*"+r+"="+i);
         }
      }
   }
}
```

```
class PerfectSquare2 {
   static int n=1000;
   public static void main(String args[]) {
      int r=1;
      do {
         System.out.println(r+"*"+r+"="+(r*r));
         r++;
      } while(r*r<n);
   }
}
```

2.1 [4 คะแนน] Big O ของโปรแกรม PerfectSquare1 เป็นเท่าใด

2.2 [4 คะแนน] Big O ของโปรแกรม PerfectSquare2 เป็นเท่าใด

รหัสนักศึกษา_____

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

3.1 [2 คะแนน] หากสั่ง System.out.println(a[a[4]-a[2]]); แล้ว จะได้ผลลัพธ์ออกมาเป็นอะไร

3.2 [2 คะแนน] หากสั่ง System.out.println(a[a[8]-a[5]]); แล้ว จะได้ผลลัพธ์ออกมาเป็นอะไร

4 จากส่วนของโปรแกรมต่อไปนี้

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

4.1 [4 คะแนน] ส่วนของโปรแกรมนี้ทำอะไร

4.2 [4 คะแนน] Big O ของส่วนของโปรแกรมนี้เป็นเท่าไหร่

รหัสนักศึกษา_____

5 จากโปรแกรมย่อยต่อไปนี้

```
static boolean methodA(Node node) {
    if( (node==null) || (node.next==null) ) return true;
    if(node.data>node.next.data) return false;
}
```

5.1 [2 คะแนน] หากกำหนดให้ Linked List มีข้อมูลเป็น root->[3]->[5]->[7]->[11]->null แล้ว โปรแกรมย่อยนี้จะ return ค่าออกมาเป็นอะไร

5.2 [4 คะแนน] โปรแกรมย่อยนี้ไว้ใช้ทำอะไร

5.3 [4 คะแนน] Big O ของโปรแกรมย่อยนี้เป็นเท่าไร

5.4 [10 คะแนน] ให้เขียนโปรแกรมย่อยนี้ใหม่โดยไม่ใช้ recursive

รหัสนักศึกษา_____

6 [4 คะแนน] การดำเนินการใดบ้างของสแตกที่ต้องใช้เวลาเป็น O(1) เสมอ

กำหนดให้มีสแตกว่างเปล่าอยู่ 2 สแตก ชื่อว่า stackA และ stackB

7 [8 คะแนน] ให้นักศึกษาเขียนข้อมูลภายในสแตก (ให้ข้อมูลขวาสุดเป็นด้านบนของสแตก) ที่ผ่านการดำเนินการต่อไปนี้

7.1 push(stackA,10)

stackA:

stackB:

7.2 push(stackB,5)

stackA:

stackB:

7.3 push(stackA,3)

stackA:

stackB:

7.4 push(stackB, pop(stackA))

stackA:

stackB:

7.5 push(stackA, pop(stackB)+pop(stackA))

stackA:

stackB:

7.6 push(stackA,7)

stackA:

stackB:

7.7 push(stackB,9)

stackA:

stackB:

7.8 push(stackA, pop(stackB)+pop(stackB))

stackA:

stackB:

รหัสนักศึกษา_____

8 [4 คะแนน] การดำเนินการใดบ้างของ queue ที่ต้องใช้เวลาเป็น O(1) เสมอ

กำหนดให้มี queue ว่างเปล่าอยู่ 2 queue ชื่อว่า queueA และ queueB

9 [8 คะแนน] ให้นักศึกษาเขียนข้อมูลภายใน queue (ให้ข้อมูลซ้ายสุดเป็นด้านหน้า queue) ที่ผ่านการดำเนินการต่อไปนี้

9.1 enqueue(queueA,7)

queueA:

queueA:

9.2 enqueue(queueB,4)

queueA:

queueA:

9.3 enqueue(queueA,6)

queueA:

queueA:

9.4 enqueue(queueB, dequeue(queueA))

queueA:

queueA:

9.5 enqueue(queueA, 1)

queueA:

queueA:

9.6 enqueue(queueB, 4)

queueA:

queueA:

9.7 enqueue(queueA, dequeue(queueA)*dequeue(queueB))

queueA:

queueA:

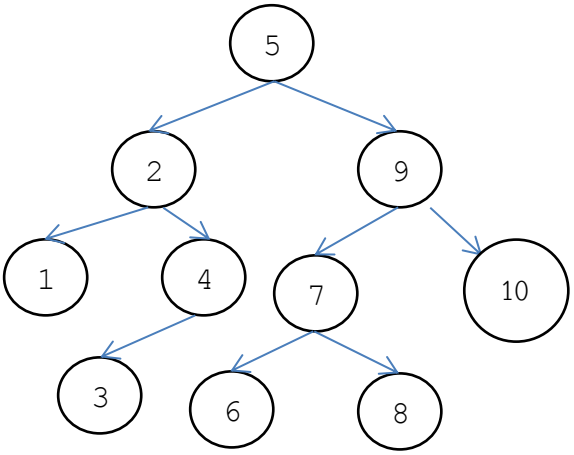
9.8 enqueue(queueB, dequeue(queueA)*dequeue(queueA))

queueA:

queueA:

รหัสนักศึกษา_____

10 [10 คะแนน] จากต้นไม้ต่อไปนี้



ให้นักศึกษาระบุ

10.1 root

10.2 path ระหว่าง 5 ถึง 6 (ไม่ต้องระบุ 5 หรือ 6)

10.3 parent ของ 7

10.4 child ของ 9

10.5 leaf

10.6 โหนดใน level 1

10.7 Height/depth

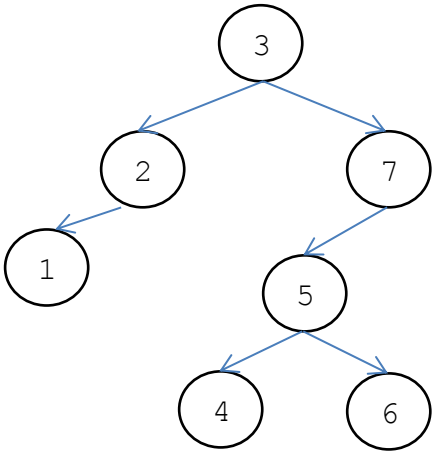
10.8 ancestor ของ 7

10.9 descendant ของ 2

10.10 subtree ของ 9

รหัสนักศึกษา_____

11 [6 คะแนน] จากต้นไม้ต่อไปนี้



ให้นักศึกษาระบุ

11.1 ให้เขียนลำดับการท่องแบบ in-order

11.2 ให้เขียนลำดับการท่องแบบ pre-order

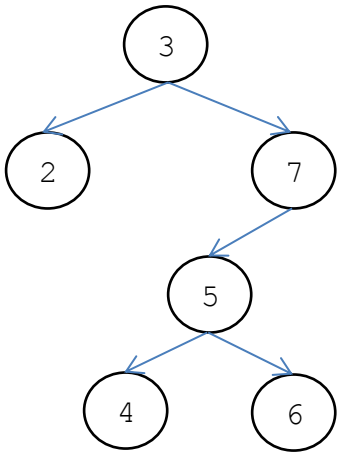
11.3 ให้เขียนลำดับการท่องแบบ post-order

12 [4 คะแนน] ให้วาดต้นไม้แบบ binary search tree ที่เกิดจากการดำเนินการต่อไปนี้

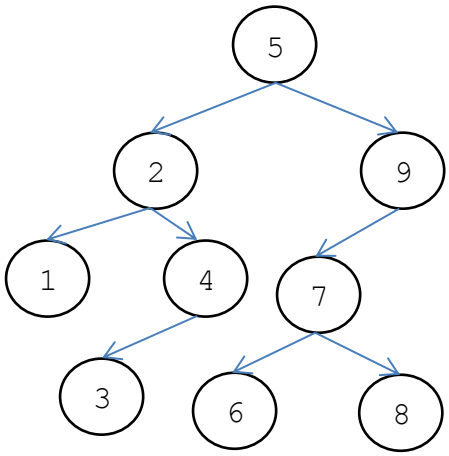
insert(15) insert(7) insert(6) insert(20) insert(5) insert(22) insert(2) insert(3) insert(21)	

รหัสนักศึกษา_____

13 [2 คะแนน] ให้วาดต้นไม้แบบ binary search tree ด้านล่างที่ผ่านการดำเนินการ delele(7) แล้ว



14 [2 คะแนน] ให้วาดต้นไม้แบบ binary search tree ด้านล่างที่ผ่านการดำเนินการ delele(5) แล้ว



รหัสนักศึกษา_____

15 [16 คะแนน] ให้นักศึกษาเขียนโปรแกรมย่อยเพื่อใส่ข้อมูลลงไปในต้นไม้ โดยมีข้อกำหนดว่า ข้อมูลในโหนดที่เป็น child ของโหนดใดๆ ภายในต้นไม้จะต้องมีค่าน้อยกว่าข้อมูลในโหนดนั้นๆ