

ชื่อ-สกุล.....รหัส.....

1. [Big O vs Benchmark] จงบอกว่า สถานการณ์ต่อไปนี้ ควรวัดประสิทธิภาพด้วย Big O หรือ Benchmark เพราะเหตุใด (ไม่มีเหตุผลประกอบไม่ได้คะแนน)

1.1 [2 คะแนน] เลือกซื้อระบบจัดการร้านเช่าการ์ตูน

1.2 [2 คะแนน] เลือกวิธีการในการค้นหาการ์ตูนเพื่อสร้างระบบร้านเช่าการ์ตูน

1.3 [2 คะแนน] เลือกวิธีจัดหนังสือการ์ตูนในร้าน

1.4 [2 คะแนน] เลือกวิธีค้นหาหนังสือการ์ตูนในร้าน

ชื่อ-สกุล.....รหัส.....

2. การคำนวณ Big O

2.1 [4 คะแนน] ออกแบบและบรรยายวิธีการในการค้นหาร้านขนมที่จะไปออกเดท และให้ระบุว่า เราใช้อะไรเป็นขนาดของปัญหา (n) และวิเคราะห์ว่า วิธีการที่เลือกใช้มี Big O เป็นเท่าใด

2.2 [4 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O(n^2 \log n)$ แล้ว ถ้าขนาดของปัญหาเพิ่มขึ้น 16 เท่า (จาก n เป็น $16n$) แล้ว ขั้นตอนวิธีนี้จะใช้เวลาเพิ่มขึ้นเป็นกี่เท่าโดยประมาณ

3. จากโปรแกรมสำหรับคำนวณ 2^n ต่อไปนี้

<pre>int exp2n1(int n) { int r=1; for (int i = 0; i < n; i++) { r = 2 * r; } return r; }</pre>	<pre>int exp2n2(int n) { int r = 1; int p = 2; while (n>0) { if (n % 2 == 1) r = r * p; p = p*p; n = n / 2; } return r; }</pre>
---	--

3.1 [4 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ exp2n1 และสรุปว่า exp2n1 มี Big O เป็นเท่าใด (ไม่แสดงวิธีทำไม่ได้คะแนน)

3.1 [6 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ exp2n2 และสรุปว่า exp2n2 มี Big O เป็นเท่าใด (ไม่แสดงวิธีทำไม่ได้คะแนน)

ชื่อ-สกุล.....รหัส.....

4. [8 คะแนน] ให้เขียนโปรแกรมเพื่อหาข้อมูลที่ซ้ำกันในอาร์เรย์ A และลิงค์ลิสต์ L โดยกำหนดให้ข้อมูลในอาร์เรย์ A เป็นข้อมูลที่จัดเรียงเสร็จแล้ว และกำหนดให้ทั้ง A และ L มีจำนวนข้อมูล (n) เท่ากัน พร้อมวิเคราะห์ Big-O

ชื่อ-สกุล.....รหัส.....

5. [8 คะแนน] กำหนดให้ list1 และ list 2 เป็นรายการโยง (linked list) ที่มีการจัดเรียงแล้วทั้งคู่ ให้เขียนโปรแกรมหาค่าของข้อมูล x ใน list1 และ y ใน list2 ที่มีค่าใกล้เคียงกันมากที่สุด พร้อมวิเคราะห์ Big O

6. จากส่วนของโปรแกรมต่อไปนี้

```

public static int main(String args[]) {
    int x[] = { 5, 3, 6, 7, 4, 9, 3, 8, 1, 2 };
    int max[] = new int[10]; int min[] = new int[10];
    min[0] = x[0];      max[9] = x[9];
    for (int i = 1; i < 10; i++) {
        if (x[i]<min[i - 1]) min[i] = x[i];
        else min[i] = min[i - 1];
    }
    for (int i = 8; i >= 0; i--) {
        if (x[i]>max[i + 1]) max[i] = x[i];
        else max[i] = max[i + 1];
    }

    int m = 0;
    for (int i = 0; i < 10; i++) {
        if (max[i] - min[i] > m) m = max[i] - min[i];
    }
    System.out.println("Answer is: " + m);
    cout << m << endl;
}

```

6.1 [6 คะแนน] อธิบายการทำงานของส่วนของโปรแกรมนี้อย่างละเอียด พร้อมทั้งแสดงผลของโปรแกรม

6.2 [6 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดส่วนของโปรแกรมนี้อย่างละเอียด พร้อมทั้งสรุปว่า Big O เป็นเท่าใด (ไม่มีวิธีทำไม่ได้คะแนน)

7. จากโปรแกรมย่อยต่อไปนี้ (กำหนด class Node ของ linked list)

```
void method7(Node p) {  
    if( (p==null) || (p.next==null) ) return;  
    p.next.data = p.next.data + p.data;  
}
```

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

7.2 [4 คะแนน] หากเรียกใช้ method7 ตามส่วนของโปรแกรมด้านล่างแล้ว ผลของ linked list จะออกมาเป็น
อย่างไร

```
Node p = head;  
while(p!=null) {  
    method7(p);  
    p = p.next;  
}
```

7.3 [4 คะแนน] Big O ของการทำงานในข้อ 7.2 เป็นเท่าใด