

1. Big O vs. Benchmark

[6 คะแนน] จงบอกว่า สถานการณ์ต่อไปนี้ ควรวัดประสิทธิภาพด้วย Big-O หรือ Benchmark เพราะเหตุใด (ไม่มีเหตุผลประกอบไม่ได้คะแนน)

1.1 เลือกซื้อระบบเพื่อใช้สำหรับลงทะเบียนให้สำนักทะเบียน

1.2 เลือกวิธีการที่จะนำมาสร้างระบบสำหรับลงทะเบียนให้สำนักทะเบียน

1.3 เลือกวิธีที่จะนำไปใช้จัดตู้หนังสือให้เป็นระเบียบ

2. การคำนวณ Big O

2.1 [2 คะแนน] หากเราต้องการคำนวณ Big O ของขั้นตอนวิธีในการค้นหาร้านอาหารที่ชอบ เราควรใช้อะไรเป็นขนาดของปัญหา (n) (อธิบายเหตุผลประกอบ)

2.2 [2 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O[\sqrt{n} \log n]$ หากขนาดของปัญหาเพิ่มขึ้นเป็น 64 เท่า (จาก n เป็น $64n$) แล้ว ขั้นตอนวิธีนี้จะใช้เวลาเพิ่มขึ้นเป็นกี่เท่าโดยประมาณ

2.3 [2 คะแนน] สำหรับขั้นตอนวิธีที่ใช้เวลาเป็น $O(2^n)$ หากขั้นตอนวิธีนี้ใช้เวลา 1 วินาที เมื่อ $n=10$ แล้ว ขั้นตอนวิธีนี้จะใช้เวลาประมาณเท่าไรเมื่อ $n=40$

3. จากโปรแกรมหา $\lceil \log(n) \rceil$ ต่อไปนี้ (จำนวนเต็มที่น้อยที่สุดที่มากกว่า $\log n$)

```
int ceilLogN1(double n) {
    int r = 1;
    int c = 1;
    while(r<n) {
        r = r * 2;
        c++;
    }
    return c;
}
```

```
int ceilLogN2(double n) {
    int r = n;
    int c = 1;
    while(r>0) {
        r = r / 2;
        ++c;
    }
    return c;
}
```

3.1 [5 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ ceilLogN1 และสรุปว่า ceilLogN1 มี Big O เป็นเท่าใด (ไม่แสดงวิธีทำไม่ได้คะแนน)

3.2 [5 คะแนน] ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดของ ceilLogN2 และสรุปว่า ceilLogN2 มี Big O เป็นเท่าใด (ไม่แสดงวิธีทำไม่ได้คะแนน)

4. [6 คะแนน] ให้เขียนโปรแกรมเพื่อหาข้อมูลที่ซ้ำกัน 1 ข้อมูลในอาร์เรย์ a และ b โดยกำหนดให้ทั้ง 2 อาร์เรย์มีขนาด n เท่ากัน พร้อมวิเคราะห์ Big-O
- ตัวอย่าง หาก $a = \{ 5, 2, 1, 0, 9, 4 \}$, $b = \{ 8, 6, 3, 4, 5, 7 \}$ จะได้คำตอบเป็น 5 หรือ 4 (ต้องการแค่คำตอบเดียว)

5. [6 คะแนน] กำหนดให้ list1 และ list2 เป็นรายการโยง (linked list) ที่มีการจัดเรียงจากน้อยไปมาก และมีขนาดเป็น n เท่ากัน ให้เขียนโปรแกรมเพื่อหาข้อมูลที่ซ้ำกัน 1 ข้อมูล พร้อมวิเคราะห์ Big-O
- ตัวอย่าง จาก list1 $\rightarrow [3] \rightarrow [6] \rightarrow [9] \rightarrow [12] \rightarrow [15] \rightarrow \text{null}$
- list1 $\rightarrow [4] \rightarrow [8] \rightarrow [12] \rightarrow [16] \rightarrow [20] \rightarrow \text{null}$ จะได้คำตอบเป็น 12 (ต้องการแค่คำตอบเดียว)

6. จากส่วนของโปรแกรมต่อไปนี้

```
int index = 0;
int b = new int[a.length];
for(int i=0; i<b.length; i++) {
    b[i]=0;
    for(int j=0; j<a.length; j++) {
        if(Math.abs(a[i]-a[j])>b)
            b[i] = Math.abs(a[i]-a[j]);
    }
}
int m=0;
for(int i=0; i<b.length; i++)
    if(m<b[i]) m = b[i];
System.out.print(m);
```

[5 คะแนน] 6.1 อธิบายการทำงานของส่วนของโปรแกรมนี้ พร้อมทั้งแสดงผลของโปรแกรม

[5 คะแนน] 6.2 ให้นับจำนวนคำสั่งของกรณีที่แย่ที่สุดส่วนของโปรแกรมนี้ พร้อมสรุปว่า Big O เป็นเท่าใด

7. จากโปรแกรมน้อยต่อไปนี้ (กำหนด class Node ของ linked list)

```
void method7(Node p) {  
    if(p==null) return;  
    Node q = p;  
    while(q.next!=null)  
        q = q.next;  
    q.next = p;  
    p.next = null;  
}
```

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 ของโปรแกรมน้อยนี้เป็นเท่าไร

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

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

8.1 [2 คะแนน] การดำเนินการใดบ้างของสแตกที่ต้องใช้เวลาเป็น $O(1)$ เสมอ

8.2 [5 คะแนน] กำหนดให้มีสแตกวางเปล่าอยู่ 2 สแตก ชื่อว่า A และ B ให้นักศึกษาเขียนข้อมูลภายในสแตก (ให้ข้อมูลล่าสุดเป็นด้านบนของสแตก) ที่ผ่านการดำเนินการต่อไปนี้

8.2.1 A.push(5) A: B:	8.2.2 B.push(4) A: B:
8.2.3 stackA.push(2) A: B:	8.2.4 B.push(A.pop()) A: B:
8.2.5 B.push(B.pop()+B.pop()) A: B:	8.2.6 A.push(3) A: B:
8.2.7 B.push(4) A: B:	8.2.8 A.push(B.pop()-A.pop()) A: B:
8.2.9 A.push(A.pop()) A: B:	8.2.10 B.push(B.top()+1) A: B:

9. จากนิพจน์ Reversed Polish Notation นี้ $1\ 2\ +\ 5\ 3\ -\ * \ 9\ 7\ 4\ -\ / \ -$ ให้วาดข้อมูลใน stack

9.1 [3 คะแนน] หลังจาก RPN Evaluation Algorithm ดำเนินการผ่านเครื่องหมายคูณแล้ว

9.2 [3 คะแนน] หลังจาก RPN Evaluation Algorithm ดำเนินการผ่านเครื่องหมายหารแล้ว

10. [10 คะแนน] โจทย์จาก uVA Online Judge ข้อ 673 กำหนดให้ input เป็นข้อมูลแบบสตริง (String) ที่มีตัวอักษรที่เป็นไปได้แค่ 4 ตัวคือ ‘(’, ‘)’, ‘[’, และ ‘]’ โดยสตริงที่นำเข้ามาจะต้อง

- 1) ถ้าเป็นสตริงเปล่า
- 2) ถ้าสตริง A และ B ถูกต้องแล้ว สตริง AB (นำมาต่อกัน) จะถูกต้องด้วย และ
- 3) ถ้าสตริง A ถูกต้องแล้ว สตริง (A) และ [A] จะถูกต้องด้วย

หรือพูดอีกนัยหนึ่งคือ สตรีงก์ที่ถูกต้องคือสตรีงก์ที่ใช้วงเล็บได้ถูกต้อง ครอบคู่ ไม่คร่อมกัน เช่น สตรีงก์

“((([])))” ผิดเพราะไม่ครบคู่ ส่วนสตริงก์ “([()[]()())” ถูกต้อง

จึงเขียนโปรแกรมเพื่อตรวจสอบสตริงว่าถูกต้องหรือไม่ โดยใช้สแตก (stack) และให้สามารถเรียกใช้คลาส Stack ได้โดยที่ไม่ต้องเขียนขึ้นมาใหม่

[illegible]