งาน

1. มานี มี arrays 2 arrays ที่จัดเก็บ ตามลำดับ คือ a_1 [1,. . . , n], a_2 [1,. . . , n + 1]

โดยใน a_1 ประกอบด้วยตัวเลขบวกที่แตกต่างกัน และ a_2 ประกอบด้วยข้อมูลที่ได้จาก a_1 แต่มีการใส่ 0 ลงใน a_2

ซึ่งมานีต้องการทราบตำแหน่ง (index) ของ 0 แต่ มานี ไม่มีเวลาเพียงพอที่จะทำการค้นหาข้อมูลใน array ดังนั้นเธอต้องการมองหาอัลกอริทึมที่รวดเร็ว ซึ่งนักศึกษาต้องออกแบบ algorithm ที่สามารถวิเคราะห์ข้อมูลได้อย่างรวดเร็วในการค้นหาดัชนีของ 0 ใน a_2 ซึ่งเวลาในการทำงาน ต้องไม่มากกว่า O (log n) มาก

เช่น . a_1 : 1, 3, 4, 6, 7, 8, 9, 20

a_2: 1, 3, 0, 4, 6, 7, 8, 9, 20.

ซึ่ง algorithm ต้องคืนตำแหน่ง ที่ 3 ของค่า 0

<u>ตอบ</u>

```
Algorithm FindZero(a2)
left ← 1
right ← length(a2)

while left ≤ right do
    mid ← (left + right) / 2

    if a2[mid] = 0 then
        return mid      // คืนตำแหน่งของ 0

    else if a2[mid] > 0 then
        right ← mid - 1
    else
        left ← mid + 1

return -1
```

2. จาก function ที่กำหนด ซึ่งเป็น function ที่ทำการแบ่ง array ออกเป็น 2 arrays ดังนั้น ให้นักศึกษาเขียนโปรแกรมในการรวม array ทั้ง 2 arrays ที่มีการจัดเรียงแล้ว

ตอบ

```
public static int[] merge(int[] left, int[] right) {
    int[] result = new int[left.length + right.length];

    int i = 0, j = 0, k = 0;

    while (i < left.length && j < right.length) {
        if (left[i] <= right[j]) {
            result[k] = left[i];
            i++;
        } else {
            result[k] = right[j];
            j++;
        }
        k++;
    }

    while (i < left.length) {
        result[k] = left[i];
        i++;
        k++;
    }

    while (j < right.length) {
        result[k] = right[j];
        j++;
        k++;
    }
```

```
    return result;
}
```

```
int middle = values.length/2; // divide array into two arrays of half size
int[] left = new int[middle];
for (int i=0; i<middle; i++) {
      left[i] = values[i];
}
int[] right = new int[values.length-middle];
for (int i=0; i<values.length-middle; i++) {
      right[i] = values[middle+i];
}
sort(left); //recursively call sorting function on each smaller array
sort(right);
```

3. จาก

A[0….7] =	15	3	9	31	11	17	7	23

3.1 แสดง insertion sort ทุกขั้นตอน

ตอบ

เริ่ม

15 3 9 31 11 17 7 23

Pass1

3 15 9 31 11 17 7 23

Pass2

3 9 15 31 11 17 7 23

Pass3

3 9 15 31 11 17 7 23

Pass4

3 9 11 15 31 17 7 23

Pass5

3 9 11 15 17 31 7 23

Pass6

3 7 9 11 15 17 31 23

Pass7

3 7 9 11 15 17 23 31

คำตอบสุดท้าย

3 7 9 11 15 17 23 31

3.2 แสดง Quick sort ทุกขั้นตอน โดย pivot คือ ตำแหน่งสุดท้าย ของ array

ตอบ

เริ่ม

15 3 9 31 11 17 7 23

Pivot = 23

แบ่งได้

15 3 9 11 17 7 |23| 31

ซ้าย

15 3 9 11 17 7

Pivot = 7

3 7 15 9 11 17

ซ้าย

3

เสร็จ

ขวา

15 9 11 17

Pivot = 17

15 9 11 17

ซ้าย

15 9 11

Pivot = 11

9 11 15

รวมทั้งหมด

3 7 9 11 15 17 23 31

4. แสดงการเพิ่มค่า 5, 28, 19, 15, 20, 33, 12, 17, 10 ลงใน ตารางแฮช hash table ที่แก้ปัญหาการชนกัน โดยการ chaining ซึ่งกำหนดให้ตารางมี 9 ช่อง และ แฮชฟังก์ชั่น (hash function) h (k) = k mod 9

ตอบ

Key	h(k) = k mod 9	Index
5	5 mod 9 = 5	5
28	28 mod 9 = 1	1
19	19 mod 9 = 1	1
15	15 mod 9 = 6	6
20	20 mod 9 = 2	2
33	33 mod 9 = 6	6
12	12 mod 9 = 3	3
17	17 mod 9 = 8	8
10	10 mod 9 = 1	1