

### ปัญหา 2.3 ตำแหน่งยอดนิยม [HotSpot] (9 คะแนน) (1.5 วินาที)

ในปัญหานี้ เราต้องการขยายการนับจำนวนค่าที่เราสนใจ ไปสู่การหาตำแหน่งที่น่าสนใจที่สุด ซึ่งก็คือ “ตำแหน่งยอดนิยม” ตำแหน่งยอดนิยมในที่นี้คือตำแหน่งที่ผลรวมของเลขที่สนใจในแถวกับคอลัมน์ของตำแหน่งนั้นมีค่ามากที่สุด สมมติว่าตำแหน่งยอดนิยมคือ แถวที่ 4 คอลัมน์ที่ 3 และเลขที่สนใจคือ 7 เราจะนับเลข 7 ที่อยู่ในแถวที่ 4 และคอลัมน์ที่ 3 ทั้งหมดมารวมกัน เช่น

0	1	7	7	3	2	3	5	4
1	7	7	2	2	7	3	0	4
0	0	0	0	0	0	0	7	2
5	7	7	5	6	7	7	8	9
4	5	7	3	1	4	2	3	7

เราจะได้ว่ามีเลข 7 รวมกันในแถวที่ 4 และคอลัมน์ที่ 3 รวมกันทั้งหมด 7 ตัว (ไม่นับเลขซ้ำ) ซึ่งสูงกว่าตำแหน่งอื่น ๆ ในตาราง ดังนั้นแถว 4 คอลัมน์ 3 จึงเป็นตำแหน่งยอดนิยม

อย่างไรก็ตาม ตำแหน่งยอดนิยมอาจจะไม่ได้มีเพียงตำแหน่งเดียวก็ได้ เป็นต้นว่า หากเราเปลี่ยนตัวเลขในการ์ดเล็กน้อยไปเป็น

0	1	7	7	3	2	3	5	4
1	7	7	2	2	7	3	7	4
0	0	0	7	0	0	0	7	2
5	7	7	5	6	7	7	8	9
4	5	7	7	1	4	2	7	7

เราจะมีตำแหน่งยอดนิยมมากถึง 6 จุดดังนี้ (1) แถว 2 คอลัมน์ 3, (2) แถว 2 คอลัมน์ 4, (3) แถว 4 คอลัมน์ 3, (4) แถว 4 คอลัมน์ 4, (5) แถว 4 คอลัมน์ 8, และ (6) แถว 5 คอลัมน์ 3

เป้าหมายของปัญหานี้ก็คือว่า เราต้องการรู้ว่า

1. ผลรวมจำนวนตัวเลขที่สนใจของตำแหน่งยอดนิยมมีกี่ตัว

2. มีตำแหน่งยอดนิยมทั้งหมดกี่จุด

3. ตำแหน่งยอดนิยมทั้งหมด

### ข้อมูลเข้า

เหมือนข้อที่แล้ว และรับประกันว่าตัวเลขที่สนใจจะปรากฏอยู่ในการ์ดอย่างน้อย 1 ใบ

### ผลลัพธ์

มีสามส่วนดังนี้

ส่วนแรก (บรรทัดที่หนึ่ง)	ผลรวมจำนวนตัวเลขที่สนใจของตำแหน่งยอดนิยม
ส่วนที่สอง (บรรทัดที่สอง)	จำนวนตำแหน่งยอดนิยมทั้งหมด
ส่วนที่สาม (มีจำนวนบรรทัดตามจำนวนตำแหน่งยอดนิยม)	ตำแหน่งยอดนิยมทั้งหมด ในรูปแบบ แถว คอลัมน์ (คั่นด้วยช่องว่าง) หนึ่งตำแหน่งต่อบรรทัด เรียงลำดับตามแถวจากน้อยไปมาก (สำหรับแถวที่ซ้ำกัน เรียงตามคอลัมน์จากน้อยไปมาก) ทั้งนี้ตำแหน่งแถวและคอลัมน์ให้เริ่มนับจาก 1 ไม่ใช่ 0

### ตัวอย่าง

ข้อมูลเข้า	ผลลัพธ์
5 9 0 1 7 7 3 2 3 5 4 1 7 7 2 2 7 3 0 4 0 0 0 0 0 0 0 7 2 5 7 7 5 6 7 7 8 9 4 5 7 3 1 4 2 3 7 7	7 1 4 3

5 9	7
0 1 7 7 3 2 3 5 4	6
1 7 7 2 2 7 3 7 4	2 3
0 0 0 7 0 0 0 7 2	2 4
5 7 7 5 6 7 7 8 9	4 3
4 5 7 7 1 4 2 7 7	4 4
7	4 8
	5 3

**คำแนะนำ 1** ถ้าเขียนเมธอดแยกออกมาให้หาผลรวมของจำนวนเลขที่สนใจในแต่ละคอลัมน์ และของแต่ละแถวเตรียมไว้ก่อน โดยหาผลรวมสำหรับคอลัมน์เก็บไว้ในอาร์เรย์อันหนึ่ง และหาผลรวมสำหรับแถวเก็บไว้ในอาร์เรย์อีกอันจะทำให้หาผลของแต่ละตำแหน่งได้ง่ายกว่าเดิม

ซึ่งการหาผลรวมสำหรับคอลัมน์นั้นก็คล้าย ๆ กับที่ทำในข้อที่ผ่านมา แต่ให้เก็บไว้ในอาร์เรย์หนึ่งมิติที่มีความยาวเท่าจำนวนคอลัมน์แทน นอกจากนี้ เราอาจจะไม่จำเป็นต้องแยกเป็นเมธอด แต่นั่นจะทำให้ main ยาวและดูเข้าใจยาก

ในกรณีที่เรารู้สึกสนใจจะเขียนเป็นเมธอดแยกออกมา ขอแนะนำให้ประกาศเมธอดเป็นดังรูปแบบข้างล่างนี้

```

static int[] sumCols(int[][] A, final int R, final int C,
                    final int target)
{
    int[] colSums = new int[C+1];

    // Enter your code here

    return colSums;
}

static int[] sumRows(int[][] A, final int R, final int C,
                    final int target)
{
    int[] rowSums = new int[R+1];

    // Enter your code here

    return rowSums;
}

```

**คำแนะนำ 2** การเก็บรายการตำแหน่งยอดนิยม ถ้าใช้อะเรย์ลิสต์แล้วค่อย ๆ add ตำแหน่งแถวและคอลัมน์เข้าไป จะทำให้โค้ดเขียนง่ายขึ้น แต่เราจะใช้อะเรย์ธรรมดาเก็บข้อมูลตำแหน่งก็ได้ ทว่าเราจะต้องคอยเก็บจำนวนตำแหน่งและเตรียมอะเรย์ที่ยาวเพียงพอไว้ด้วย

สำหรับคนที่อยากจะใช้อะเรย์ลิสต์ ขอแนะนำให้ประกาศการเก็บ “วัตถุจำนวนเต็ม” ด้วยโค้ดแบบข้างล่างนี้ ซึ่งวัตถุจำนวนเต็มที่ว่านี้ เวลาเราจะใช้งานมันแบบจำนวนเต็มปกติ (**primitive type**) เรายังใช้งานมันได้ในลักษณะเดิมที่เราคุ้นเคย

```

ArrayList<Integer> maxRows = new ArrayList<>();
ArrayList<Integer> maxCols = new ArrayList<>();

```

ตอนที่ใส่ข้อมูลเข้าไป สามารถเขียนได้ว่า

```

maxRows.add(row);
maxCols.add(col);

```

ส่วนตอนที่ลบค่าในอาร์เรย์ลิสต์ทั้งหมด ในที่นี้เราอาจจะเขียนว่า

```
maxRows.removeAll(maxRows);
```

```
maxCols.removeAll(maxCols);
```