

Moving Mode

Mode (ฐานนิยม) คือค่าทางสถิติที่ระบุถึงค่าที่ปรากฏบ่อยที่สุดในข้อมูลของเรา โจทย์ข้อนี้จะพิจารณาถึงการวิเคราะห์ข้อมูลแบบหนึ่งที่เรียกว่า moving mode ขนาด k ซึ่งเป็นวิธีการวิเคราะห์ข้อมูลของข้อมูลตามแกนเวลาแบบหนึ่ง เช่น สมมติให้เรามีข้อมูลอุณหภูมิรายวันอยู่ โดยให้ $t[i]$ คืออุณหภูมิของวันที่ i เราจะนิยามค่า moving mode ขนาด k ของวันที่ j ใด ๆ คือ ค่าที่ปรากฏบ่อยครั้งที่สุดจากกลุ่มข้อมูล $t[j-k], t[j-k+1], t[j-k+2], \dots, t[j-1], t[j], t[j+1], \dots, t[j+k-2], t[j+k-1], t[j+k]$ กล่าวคือ ค่า moving mode ของวันที่ j จะพิจารณาจากค่าของวันที่ $j-k$ ถึงวันที่ $j+k$ นั่นเอง

เรามีข้อมูลอุณหภูมิของวันต่าง ๆ บางวัน (อาจจะไม่ครบทุกวัน) ของวันที่ 1 ถึงวันที่ n จงคำนวณค่า moving mode ขนาด k ของแต่ละวัน เพื่อความชัดเจน ขอกำหนดเพิ่มเติมดังนี้

- หากมีข้อมูลที่ปรากฏบ่อยสุดมากกว่า 1 ค่าเท่ากัน เราจะเลือกข้อมูลที่มีค่าน้อยกว่าให้เป็นค่า moving mode ตัวอย่างเช่น หากกลุ่มข้อมูลมีค่าเป็น 5, 4, 5, 7, 4 เราจะถือว่าค่า moving mode คือ 4 เพราะถึงแม้ทั้ง 4 และ 5 จะปรากฏบ่อยครั้งที่สุดคือสองครั้งเท่ากัน แต่ 4 มีค่าน้อยกว่า
- ในการคำนวณค่า moving mode ของวันที่ j นั้น หากมีวันใดในช่วงวันที่ $j-k$ ถึง $j+k$ ไม่มีข้อมูล เราจะไม่นำข้อมูลดังกล่าวมาเป็นส่วนหนึ่งของการคำนวณ นอกจากนี้ ในกรณีที่วันที่ $j-k$ ถึง $j+k$ ไม่มีข้อมูลเลยแม้แต่วันเดียว ให้ถือว่า ค่า moving mode ของวันที่ j จะไม่สามารถคำนวณค่าได้เนื่องจากข้อมูลไม่เพียงพอ

ตารางข้างล่างนี้แสดงถึงการคำนวณค่า moving mode ขนาด 2 ของข้อมูลจำนวน 10 วัน (ช่องที่เว้นว่างไว้หมายถึงไม่มีข้อมูลของวันดังกล่าว หรือหมายถึงไม่สามารถคำนวณค่า moving mode ของวันดังกล่าวได้เนื่องจากข้อมูลไม่เพียงพอ)

วันที่	1	2	3	4	5	6	7	8	9	10
ค่า $t[i]$	7	7	8	9	8		2			
ค่า moving mode	7	7	7	8	8	2	2	2	2	

- ให้สังเกตว่า ค่า moving mode ของวันที่ 1 คือ 7 เนื่องจาก เมื่อพิจารณาตั้งแต่วันที่ 1-2 ถึง 1+2 จะมีข้อมูลอยู่ 3 วันคือ 7, 7, 8 ซึ่งค่าที่ปรากฏบ่อยสุดคือ 7
- ให้สังเกตว่า ค่า moving mode ของวันที่ 3 คือ 7 เนื่องจาก เมื่อพิจารณาตั้งแต่วันที่ 3-2 ถึง 3+2 จะมีข้อมูลอยู่ 5 วันคือ 7, 7, 8, 9, 8 ซึ่งค่าที่ปรากฏบ่อยสุดคือ 7 และ 8 ซึ่งปรากฏสองครั้งทั้งคู่ เนื่องจากค่า 7 มีค่าน้อยกว่า เราจึงเลือกค่า 7
- ให้สังเกตว่า ค่า moving mode ของวันที่ 7 คือ 2 เนื่องจาก เมื่อพิจารณาตั้งแต่วันที่ 7-2 ถึง 7+2 จะมีข้อมูลอยู่ 2 วันคือ 2 และ 8 เราจึงเลือกค่า 2
- ให้สังเกตว่า ค่า moving mode ของวันที่ 8 คือ 2 เนื่องจาก เมื่อพิจารณาตั้งแต่วันที่ 8-2 ถึง 8+2 จะมีข้อมูลอยู่ 1 วันคือ 2 เราจึงเลือกค่า 2

จงเขียนโปรแกรมเพื่อคำนวณค่า moving mode ขนาด k ของวันที่ 1 ถึงวันที่ N จากข้อมูลที่กำหนดให้

ข้อมูลนำเข้า

บรรทัดแรกประกอบด้วยจำนวนเต็มสามตัวคือ k และ N และ M (รับประกันว่า $0 \leq k \leq 10$ และ $1 \leq M \leq N \leq 10^5$)

อีก M บรรทัดถัดมาจะเป็นข้อมูลอุณหภูมิของแต่ละวัน บรรทัดละ 1 วัน โดยในแต่ละบรรทัดประกอบด้วยจำนวนเต็ม 2 ตัวคือ a และ b ซึ่งระบุว่าวันที่ a มีค่าอุณหภูมิเป็น b นาโนเซลเซียส (รับประกันว่า $1 \leq a \leq N$ และ $-10^9 \leq b \leq 10^9$)

ข้อมูลส่งออก

มีทั้งหมด 1 บรรทัด ประกอบด้วยตัวเลข N ตัวที่ระบุค่า moving mode ของวันที่ 1 ถึง N ตามลำดับ หากวันใดไม่สามารถคำนวณค่า moving mode ได้ ให้แสดงตัวอักษร X (ตัวพิมพ์ใหญ่) แทนตัวเลข

ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)	คำอธิบาย
2 10 6 1 7 2 7 3 8 4 9 5 8 7 2	7 7 7 8 8 2 2 2 2 X	ตัวอย่างนี้ตรงกับตัวอย่างที่แสดงในตารางด้านบน
10 3 3 2 10 3 10 1 5	10 10 10	
0 10 2 7 1 3 2	X X 2 X X X 1 X X X	

ชุดข้อมูลทดสอบ

ข้อมูลต่อไปนี้อธิบายว่าข้อมูลทดสอบของโจทย์ข้อนี้มีลักษณะอย่างไรบ้าง นิสิตสามารถใช้ข้อมูลเหล่านี้ช่วยในการทำโจทย์ได้

- 5% รับประกันว่า $k = 0$
- 10% รับประกันว่า $k = 1$ และ $N = M$ (กล่าวคือ มีข้อมูลของอุณหภูมิของทุกวัน) และ ข้อมูลอุณหภูมิที่ให้เรียงจากน้อยไปมาก (กล่าวคือ ในแต่ละบรรทัด ค่า a จะเพิ่มขึ้นทีละ 1 และค่า a แรกในบรรทัดที่ 2 จะเป็น 1)
- 5% รับประกันว่า $k = 1$ และ $N = M$
- 20% รับประกันว่า $k \leq 3$
- 20% รับประกันว่า $a \leq 10^5$
- 30% ไม่มีเงื่อนไขพิเศษอื่นใด

```

#include <iostream>
#include <vector>
#include <map>

using namespace std;
int k,N,M;

int main() {
    cin >> k >> N >> M;
    map<int,int> data;
    for (int i = 0;i < M;i++) {
        int a,b;
        cin >> a >> b;
        data[a] = b;
    }

    //for each day we have to answer
    map<int,int> freq; //freq[i] = frequency of i in the windows

    //build initial freq of day 1..k-1
    for (int i = 1;i <= k;i++) {
        if (data.find(i) != data.end()) {
            freq[data[i]]++;
        }
    }
    for (int j = 1;j <= N;j++) {
        //count new data
        int new_data_day = j+k;
        if (data.find(new_data_day) != data.end()) {
            freq[data[new_data_day]]++;
        }

        //remove old data
        int old_data_day = j-k-1;
        if (data.find(old_data_day) != data.end()) {
            freq[data[old_data_day]]--;
            if (freq[data[old_data_day]] == 0) {
                freq.erase(data[old_data_day]);
            }
        }

        //find max freq
        if (freq.size() == 0) {
            cout << "X ";
        } else {
            int max = 0, max_freq = 0;
            for (auto &x : freq) {
                if (x.second > max_freq) {
                    max_freq = x.second;
                    max = x.first;
                }
            }
            cout << max << " ";
        }
    }
}

```