2024 NCKU Program Design I Homework 5

> 請不要嘗試攻擊 Judge 系統，否則此堂課將不予以通過
>
> Don't attack judge system otherwise you will fail this course.
>
> 一旦發現作業抄襲或是請人代寫之情形，學期作業成績將 "全部" 以 0 分計算
>
> One instances of severe plagiarism, hiring someone to write assignments, or similar activities are detected, the semester's assignment scores will be calculated as 0 point across the board.

> 對於作業有任何問題可以直接寄信到 f74114744@gs.ncku.edu.tw (mailto:f74114744@gs.ncku.edu.tw)，如果是其他課程上的問題需要處理請利用全體助教信箱 c2024@mail.csie.ncku.edu.tw (mailto:c2024@mail.csie.ncku.edu.tw)
>
> If you have any question about this homework tasks (ex. problem description), please feel free to contact me (資訊 115 陳俊安, f74114744@gs.ncku.edu.tw (mailto:f74114744@gs.ncku.edu.tw)).
>
> Otherwise, the common problem of this homework please send to TAs mail (c2024@mail.csie.ncku.edu.tw (mailto:c2024@mail.csie.ncku.edu.tw))

Homework 5 Information

Deadline: 11/20 23:59:59

> 不接受遲交
>
> Late submissions are not accepted.

Before you start

- Make sure you can login the server by your personal account.

Goals

- Arrays
- Function

Submission

- Server IP: `140.116.246.48` , Port: `2024`

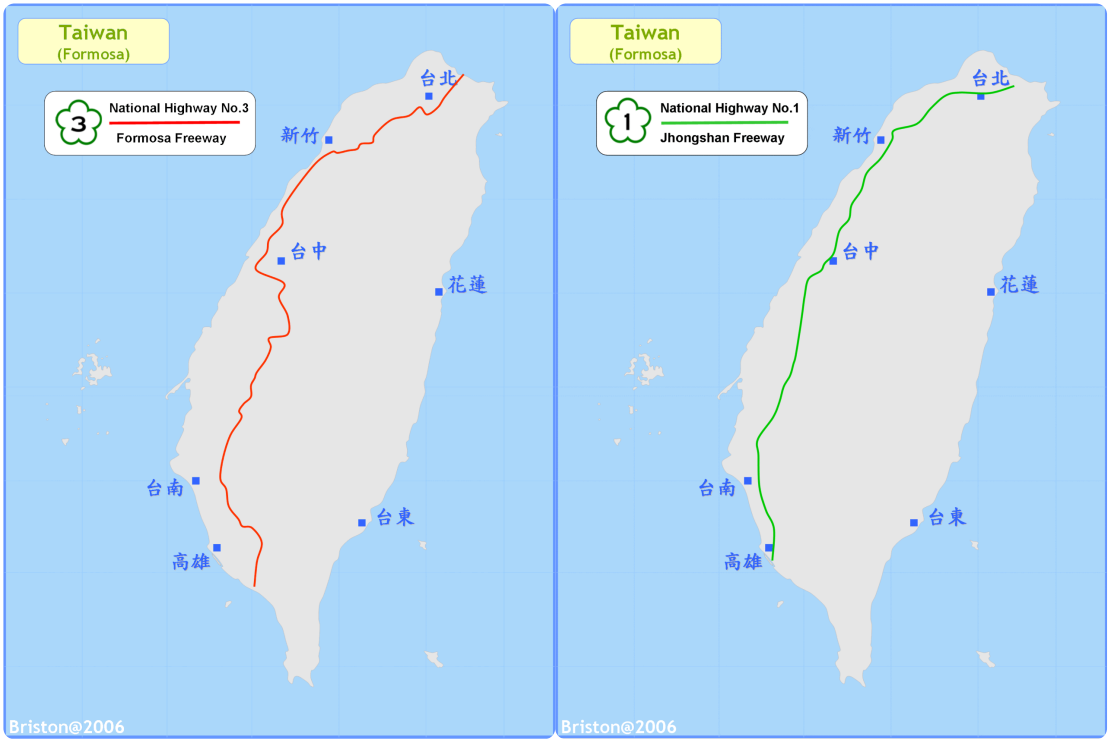
```
ssh 學號@140.116.246.48 -p 2024
```

- Login the system by your personal account. (Use the ssh command)
- Create an directory with name `HW5` in your home directory.
  - You can use the "pwd" command to confirm your current directory.
  - The "mkdir [name]" command can create a directory with the name [name]
- In HW5 directory, you need to create 8 files with name "pA.c", "pB.c", "pC.c", "pD.c", "pE.c", "pF.c", "pG.c" and "pH.c" respectively.
- You can directly use the command `hw5` to check whether the result of each question is correct.

> The command is not available at the moment. We will update it as soon as possible.

Tasks

Problem A. The Trip to Sun Moon Lake (pA.c) - 10%



Colten often drives on day trips to magical places, and one day, he traveled to Sun Moon Lake.

On the way to Sun Moon Lake, he drove on National Freeways No.1 and No.3 (國道一號、三號). There are ramps (閘道, gateways) on these freeways, and these ramps usually have large turns, so the speed limit on these ramps is typically between 40 and 60 km/h.

Let's explore a cool question: we have obtained a set of speed data from the car, recording the actual speed over a continuous period (a total of N speed records).

Now, we want to know how many times the vehicle may have passed through a ramp in this recorded data. Please design a program to calculate this.

> **Formal Problem Definition:**
>
> You are given N numbers $s_1, s_2, \ldots, s_N$. We want to determine if there exists M pairs $(L_1, R_1), (L_2, R_2), \ldots, (L_M, R_M)$ that satisfy the following conditions:
>
> 1. **Condition 1**:
>   For each pair (L_i, R_i) where $i \in [1, M]$, every $j \in [L_i, R_i]$ satisfies:
> $$40 \leq s_j \leq 60$$
> 2. **Condition 2**:
>   Define $S_k = \{L_k, L_k + 1, \ldots, R_k\}$ for each $k \in [1, M]$.
>   Then, for any index t where $40 \leq s_t \leq 60$, it must hold that:
> $$t \in (S_1 \cup S_2 \cup \ldots \cup S_M)$$
>
>   In addition to this, for any index x where $60 < s_x$, it must hold thst:
> $$x \notin (S_1 \cup S_2 \cup \ldots \cup S_M)$$
>
> In other words, any value s_t in the range $[40, 60]$ must be part of any defined sets S_k and any value s_x in the range $[61, 110]$ must be not a part of any defined sets S_k 。
>
> You need to write a program to determine the minimum possible value of M。

Fun Fact

> National Freeway No. 1, also known as the Sun Yat-sen Freeway (中山高速公路)
>
> National Freeway No. 3, also known as the Formosa Freeway (福爾摩沙高速公路)

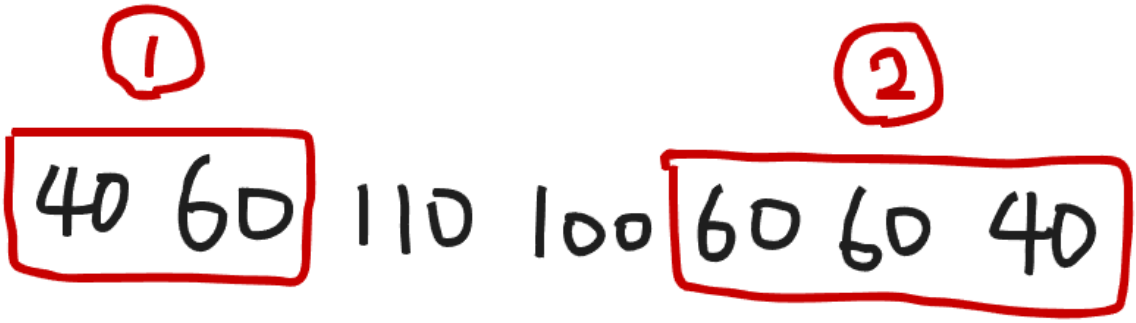
Input Format

```
N
s_1 s_2 ,..., s_N
```

- $1 \leq N \leq 2 \times 10^5$
- $40 \leq s_i \leq 110$

Sample Input 1

```
7
40 60 110 100 60 60 40
```



Sample Output 1

```
2
```

Sample Input 2

```
1
59
```

Sample Output 2

```
1
```

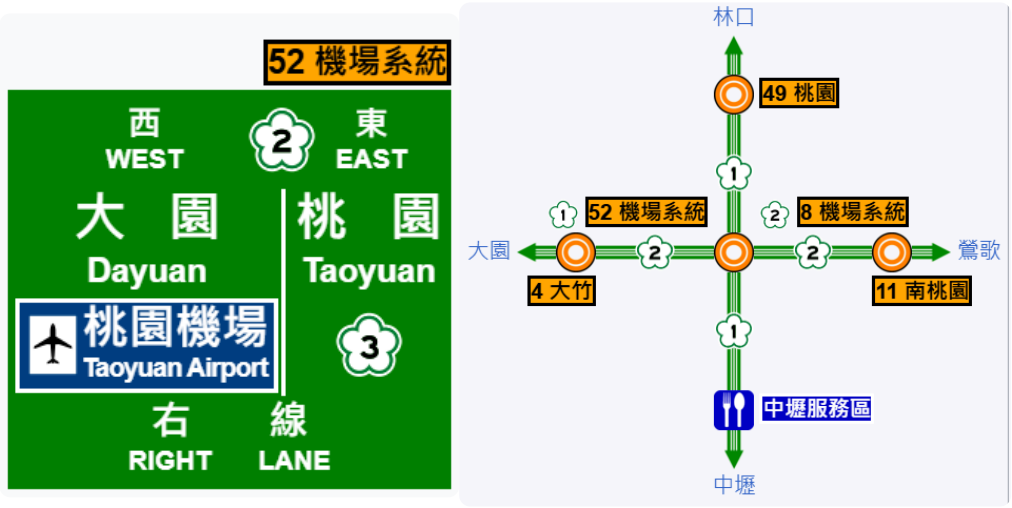
Sample Input 3

```
12
40 60 60 60 40 100 40 100 60 60 100 110
```

Sample Output 3

```
3
```

Problem B. Airport System Interchange in National Freeway No. 1 [Easy Version] (pB.c) - 10%



> This problem has a easy version and a hard version, with the difference between them being the range of L_i, R_i.
>
> In the easy version, L_i, R_i goes up to a maximum of 1000.

Taiwan Taoyuan International Airport is the busiest airport in Taiwan in terms of passenger traffic, and the **Airport System Interchange** connecting the airport terminals has become a primary transportation artery.

Today, we want to take a closer look at the number of vehicles simultaneously present in the **Airport System Interchange** during peak hours.

To simplify the problem, we only need to find the maximum number of vehicles present simultaneously within the **Airport System Interchange**.

Given N vehicles with entry times L_i and exit times R_i, please design a program to calculate the answer.

Note specifically that if a vehicle exits at time R_i, it is not considered present at time R_i.

Input Format

```
N
L_1 R_1
L_2 R_2
.
.
.
L_N R_N
```

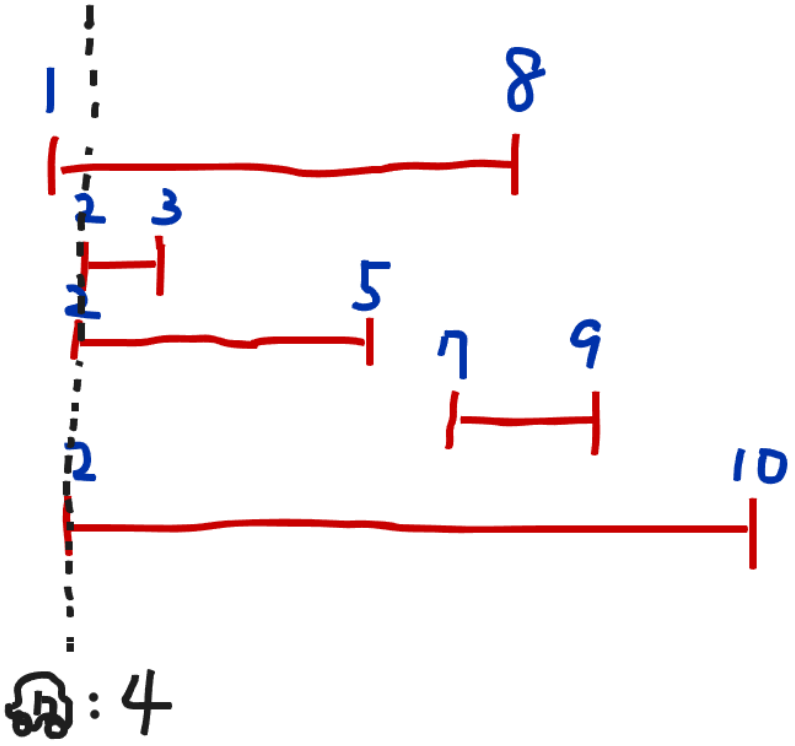
- $1 \leq N \leq 5000$
- $1 \leq L_i \leq R_i \leq 1000$

Sample Input 1

```
5
1 8
2 3
2 5
7 9
2 10
```

Sample Output 1

```
4
```



Problem C. Airport System Interchange in National Freeway No. 1 [Hard Version] (pC.c) - 10%

> This problem has a easy version and a hard version, with the difference between them being the range of L_i, R_i.
>
> In the easy version, L_i, R_i goes up to a maximum of 10^6.

The requirements for this problem are the same as the previous one.
However, here we'll look at a smarter approach to solve it.

We define an array $d[i]$ to represent the change in the number of vehicles at each time point i.

When a vehicle enters at L_i and exits at R_i, it implies that the change at $d[L_i]$ should increase by 1, and the change at $d[R_i]$ should decrease by 1.

With this, we have the change in vehicle count at each time point. Next, we simply use a variable to keep track of the total number of vehicles, adding the change at each time point to this variable. This will give us the vehicle count at every time point, and our desired answer is the maximum value among these counts.

This technique is called the **"sweep line" (掃描線)** approach.

Input Format

```
N
L_1 R_1
L_2 R_2
.
.
.
L_N R_N
```

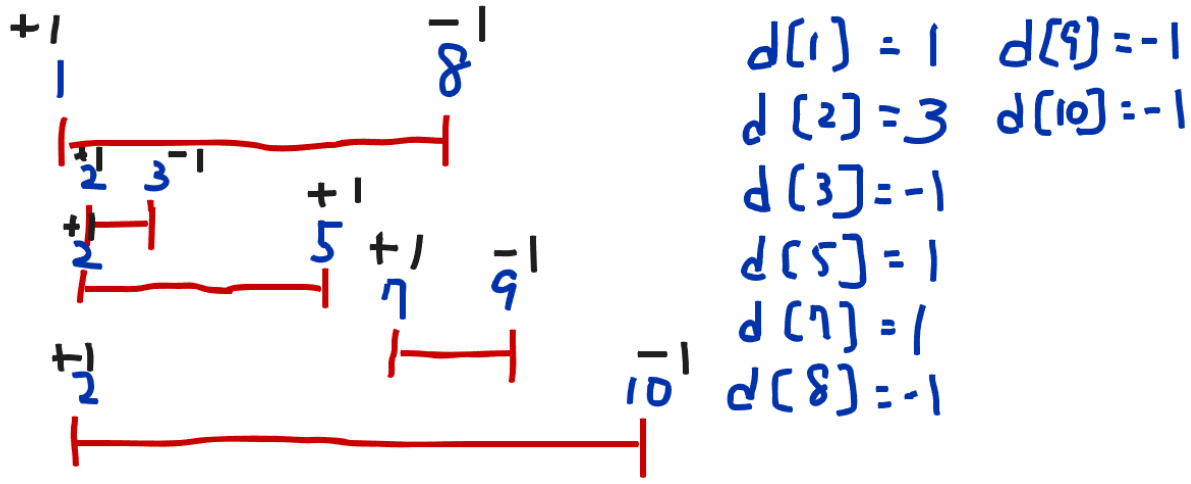
- $1 \leq N \leq 5000$
- $1 \leq L_i \leq R_i \leq 10^6$

Sample Input 1

```
5
1 8
2 3
2 5
7 9
2 10
```

Sample Output 1

```
4
```



Problem D. Range Queries of Sum (pD.c) - 15%

Given a sequence and some queries that require two positive integers to find the sum of a given range.

> Using a loop to calculate the sum for each query every time is a highly inefficient approach.
>
> This is actually an educational question, and the technique for this question is called **"prefix sum"**.
>
> We can create an additional array and define it: $b[i] = a_1 +, \ldots, + a_i$
>
> **Please build the array `b` before querying the range sum**
>
> $[L, R] = a[L] + \ldots + a[R]$
> $b[R] = a[1] + \ldots + a[L-1] + a[L] + \ldots + a[R]$
> $b[L-1] = a[1] + \ldots + a[L-1]$
> $b[R] - b[L-1] = a[L] + \ldots + a[R]$
>
> This way, you don't need to calculate anything with a loop for each query, which greatly improves efficiency!

Input Format

```
N (Length of sequence)
a_1 ,..., a_N
Q (Query Times)
L_1 R_1
L_2 R_2
.
.
.
L_Q R_Q
```

- $1 \leq N, Q \leq 2 \times 10^5$
- $1 \leq L_i \leq R_i \leq N$
- $1 \leq a_i \leq 10^9$

Sample Input 1

```
5
1 2 3 4 5
2
1 5
3 4
```

Sample Output 1

```
15
7
```

Problem E. High-Occupancy Vehicle Restrictions [Easy Version] (pE.c) - 15%

> This problem has a easy version and a hard version, with the difference between them being the range of n, q.
>
> In the easy version, n, q goes up to a maximum of 100.

High-occupancy vehicle (HOV) restrictions are a method to reduce traffic congestion by requiring a minimum number of passengers in a vehicle for it to drive on designated routes. This rule is commonly implemented on highways in Taiwan.

There are n vehicles that have passed through, with each vehicle having p_i passengers. The required number of passengers under today's HOV restrictions is k.
You are tasked with designing a program to handle q queries. Each query consists of two positive integers $[l, r]$, representing a range of vehicles.

For each query, you need to determine how many vehicles within the specified range violate the HOV restriction.

Please write a program to solve them.

Input Format

```
n q k
p_1 ... p_n
l_1 r_1
.
.
.
l_n r_n
```

- $1 \leq n, q \leq 100$
- $1 \leq k, p_i \leq 10^9$
- $1 \leq l_i \leq r_i \leq n$

Sample Input 1

```
5 3 3
1 2 3 4 5
1 1
1 5
2 4
```

Sample Output 1

```
1
2
1
```

Sample Input 2

```
5 2 10
9 102 199 4 3
1 5
3 5
```

Sample Output 2

```
3
2
```

Problem F. High-Occupancy Vehicle Restrictions [Hard Version] (pF.c) - 15%

> This problem has a easy version and a hard version, with the difference between them being the range of n, q.
>
> In the easy version, n, q goes up to a maximum of 10^5.

The only difference in this problem compared to the previous one is the range of n and q, requiring us to use the techniques from `Problem D` to solve it efficiently.

We can start by marking vehicles that violate the HOV restriction as `1` and those that do not as `0`.

The problem then simplifies to finding the count of 1s within each query range. This can be efficiently achieved by using the `Prefix Sum` technique from `Problem D`.

Input Format

```
n q k
p_1 ... p_n
l_1 r_1
.
.
.
l_n r_n
```

- $1 \leq n, q \leq 10^5$
- $1 \leq k, p_i \leq 10^9$
- $1 \leq l_i \leq r_i \leq n$

Sample Input 1

```
5 3 3
1 2 3 4 5
1 1
1 5
2 4
```

Sample Output 1

```
1
2
1
```

Sample Input 2

```
5 2 10
9 102 199 4 3
1 5
3 5
```

Sample Output 2

```
3
2
```

Problem G. Special Formula (pG.c) - 10%

Given three numbers a, b, and c, you can substitute them in any order into a function $f(x, y, z)$:

$$f(x, y, z) = ((x + z) \oplus (y - z)) - (x \wedge z \wedge (y \wedge 2024) \vee (|z - y|))$$

Please design a program to compute the maximum value of $f(x, y, z)$ when a, b, and c are substituted in any possible order.

Input Format

```
a b c
```

$1 \leq a, b, c \leq 1000$

Sample Input 1

```
10 20 30
```

Sample Output 1

```
24
```

Sample Input 2

```
1000 1 1
```

Sample Output 2

```
1001
```

Problem H. City Skyline (pH.c) - 15%

You are given a two-dimensional grid ($n \times m$) representing the height of buildings in a city ($h_{i,j}$). Each cell in the grid contains a non-negative integer that represents the height of a building at that position. Your goal is to maximize the city's skyline by increasing building heights without altering the city's skyline from the north, south, east, or west perspectives.

Skyline View:

- The `north-south skyline` is the **maximum height of buildings in each column**.
- The `east-west skyline` is the **maximum height of buildings in each row**.

Rules:
You may increase the height of buildings but cannot decrease them.
The skyline from the north, south, east, and west should remain the same as the original.

Task:
Write a program to calculate the maximum sum of height increases for all buildings while preserving the original skyline from each direction.

Input Format

```
n m
h_{1,1} ... h_{1,m}
.
.
.
h_{n,1} ... h_{n,m}
```

- $1 \leq n, m \leq 2000$
- $1 \leq h_{i,j} \leq 5000$

Sample Input 1

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

Sample Output 1

```
35
```