PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Diseño y Análisis de Algoritmos - IIC2283
Tarea 1
Entrega: 16 de septiembre (hasta las 23:59pm)

La solución de cada problema debe ser un programa en Python 3. Estos programas deben ser entregadas a través del Siding. El formato de entrega será informado en el foro del curso en el Siding. En este foro además puede hacer preguntas sobre los problemas de la tarea. Si quiere usar alguna librería en sus soluciones debe preguntar en el foro del curso si esta librería está permitida. El foro es el canal de comunicación oficial para todas las tareas.

Problema 1

Un maestro tiene que pintar una cerca. La cerca consiste en una serie de tablas rectangulares verticales, donde cada tabla tiene ancho 1 y la $i-$ésima tabla tiene altura h_i. El maestro cuenta con una brocha de ancho 1, la que le permite pintar a base de *trazos*. Se entiende por *trazo* la acción de sujetar la brocha verticalmente y deslizarla horizontalmente sobre la cerca, o bien sujetar la brocha horizontalmente y deslizarla verticalmente sobre la cerca, de tal manera que en todo momento la totalidad de la brocha siempre esté en contacto con la cerca (la brocha no puede estar parcialmente dentro y parcialmente en el aire, y si en algún instante la brocha se despega de la cerca entonces ahí termina el trazo). ¿Cuál es la mínima cantidad de trazos que el maestro debe realizar para pintar la cerca completamente?

Formato

El input consistirá de dos líneas. La primera línea es un entero N indicando la cantidad de tablas de la cerca.
Luego vienen N enteros separados por un espacio indicando las alturas de las tablas:
N
$h_1\ h_2\ \cdots\ h_n$
Su output debe ser una única línea con la mínima cantidad de *trazos* necesarios para pintar la cerca.

Límites

$1 \leq N \leq 5000$
$1 \leq h_i \leq 10^9$ para cada $i \in [1, N]$

Tiempo de ejecución

5 segundos

Complejidad esperada

$O(N^2)$

Ejemplos

Input 1

```
5
2 2 1 2 1
```

Output 1

```
3
```

Input 2

```
2
2 2
```

Output 2

```
2
```

Input 3

```
1
5
```

Output 3

```
1
```

Problema 2

Se tiene un arreglo de enteros $A = [a_1, \ldots, a_n]$. Consideramos que un subarreglo S de A es válido si $S = [a_{k_1}, \ldots, a_{k_m}]$, con $k_i \in [1, n]$ y $k_i + 2 < k_{i+1}$ para cada $i \in [1, m-1]$. Definimos $f(S) = \sum_{i=1}^{m} a_{k_i}$. Su programa debe recibir un arreglo A y devolver:

$$M \ = \ \text{máx}\{f(S) \mid S \text{ es un subarreglo válido de } A\}.$$

Formato

El input consistirá de dos líneas de números separados por espacios:

n

$a_1 \ a_2 \ \cdots \ a_n$

Su output debe ser una única línea con el valor M.

Límites

$1 \leq n \leq 10^6$

$0 \leq a_i \leq 10^9$ para cada $i \in [1, n]$

Tiempo de ejecución

5 segundos

Complejidad esperada

$O(n)$

Ejemplos

Input 1

```
2
4 5
```

Output 1

```
5
```

Input 2

```
10
61 72 24 12 61 7 59 47 46 51
```

Output 2

```
184
```

Este resultado se obtiene con el subarreglo `[72, 61, 51]`. Cualquier otro subarreglo válido da una suma menor como resultado.

Problema 3

Se tiene dos arreglos de puntos con coordenadas enteras $A = [(x_1^A, y_1^A), \ldots, (x_n^A, y_n^A)]$ y $B = [(x_1^B, y_1^B), \ldots, (x_n^B, y_n^B)]$ que cumplen con las siguientes propiedades:

- $x_i^A < x_{i+1}^A$ e $y_i^A > y_{i+1}^A$ para cada $i \in [1, n-1]$.
- $x_i^B < x_{i+1}^B$ e $y_i^B > y_{i+1}^B$ para cada $i \in [1, n-1]$.
- $y_1^A < y_n^B$ y $x_n^A < x_1^B$

Definimos $f : [1, n] \times [1, n] \to \mathbb{N}$ como:

$$f(i, j) \;=\; (x_j^B - x_i^A)(y_j^B - y_i^A).$$

Es decir, $f(i, j)$ es el área del rectángulo que forman los puntos (x_i^A, y_i^A) y (x_j^B, y_j^B). Su programa debe recibir estos dos arreglos y devolver

$$M = \text{máx}\{f(i, j) \mid 1 \leq i, j \leq n\}.$$

Formato

El input consistirá de tres líneas de números separados por espacios:

n
$x_1^A \; y_1^A \; x_2^A \; y_2^A \; \cdots \; x_n^A \; y_n^A$
$x_1^B \; y_1^B \; x_2^B \; y_2^B \; \cdots \; x_n^B \; y_n^B$

Su output debe ser una única línea con el valor M.

Límites

$1 \leq n \leq 10^5$
$-10^9 \leq x_i^A, y_i^A \leq 10^9$ para cada $i \in [1, n]$
$-10^9 \leq x_i^B, y_i^B \leq 10^9$ para cada $i \in [1, n]$

Tiempo de ejecución

5 segundos

Complejidad esperada

$O(n \log n)$

Ejemplos

Input 1

```
2
0 1 1 0
3 3 5 2
```

Output 1

```
8
```

Input 2

```
3
-3 4 -1 1 3 -1
4 9 6 8 7 7
```

Output 2

```
49
```

Problema 4

Se tiene un número entero positivo N. Se dice que un entero positivo M es *amigo* de N si y sólo si:

1. Los dígitos de M en base 10 son una permutación de los dígitos de N en base 10 (Nota: el primer dígito de un número positivo no puede ser 0, por ejemplo si $N = 100$, las permutaciones 001 y 010 no son válidas).
2. M es un múltiplo de 11.

Su programa debe recibir un entero positivo N y devolver:

$$K(N) \;=\; |\{M \in \mathbb{N} \mid M \text{ es } \textit{amigo} \text{ de } N\}| \mod (10^9 + 7)$$

Es decir, la cantidad de enteros positivos M que son *amigos* de N, pero dado que dicha cantidad puede ser muy grande, se pide el resto de dividir esa cantidad por $10^9 + 7$.

Formato

El input consistirá de una sola línea con un entero positivo N. Su output debe ser una única línea con la cantidad de *amigos* de N mod $10^9 + 7$.

Límites

$1 \leq N \leq 10^{100}$

Tiempo de ejecución

5 segundos

Complejidad esperada

$O(\log_{10}(N)^3)$

Ejemplos

Input 1

```
2090
```

Output 1

```
2
```

Para $N = 2090$ las permutaciones válidas de sus dígitos son 2009, 2090, 2900, 9002, 9020 y 9200, de las cuales sólo 2090 y 9020 satisfacen la condición de ser múltiplo de 11, así que la respuesta es 2.

Input 2

```
73251455528701
```

Output 2

```
37592100
```