Defensa a la muralla (muralla)

Descripción

En el antiguo y nada fantasioso reino de **Obilan**, existía un conflicto con el reino vecino **Trusia**, pues desde hace generaciones Trusia se propuso realizar un ataque al reino de Obilan, y finalmente ha llegado el trágico día. Ambos reinos se encuentran separados por una muralla de **N columnas**, donde cada columna tiene un ancho de 1 y una altura definida por el valor `h_i`.

La princesa **Nathalie**, encargada de la seguridad e integridad del reino de Obilan, realizó una jugada que podría garantizar la victoria en la defensa de la muralla. Gracias a sus espías en Trusia, Nathalie sabe que el reino enemigo realizará **Q ataques**.

Cada ataque se llevará a cabo en una porción de la muralla: el **ataque** `Q_i` se realizará entre las columnas `l_i` y `r_i`, y en este ataque Trusia intentará derribar `w_i` columnas consecutivas dentro de ese rango. Es decir, el ataque puede realizarse en cualquier porción de `w_i` columnas consecutivas que pertenezcan al intervalo `[l_i, r_i]`.

Dado que Nathalie conoce a la perfección la muralla, quiere reforzar una porción de cada rango que Trusia atacará. Para esto, reforzará un **cuadrilátero de ancho** `w_i` **y altura** `H` en el rango del ataque `Q_i`. este cuadrilatero, debe caber sin sobrepasar la altura de alguna columna de la porcion que se escoja. Sin embargo, Nathalie no sabe exactamente qué altura elegir en cada rango de ataque.

Tu tarea es ayudarla: dado un rango de ataque `Q_i` y el ancho `w_i` del cuadrilátero, determina la **altura máxima `H` ** que Nathalie puede escoger para dicho cuadrilátero.

Implementación

Debes implementar la siguiente función en C++:

```
std::vector<int> altura_maxima(
  int N, std::vector<int> columnas, int Q, std::vector<std::tuple<int,
)
```

O la siguiente función en Python:

```
def altura_maxima(
  N: int, columnas: list[], Q: int, ataques: list[tuple[int,int,int]]
) -> list[int]:
```

Parámetros

- `N`: que representa el número de columnas de la muralla.
- `columnas`: Un arreglo de N enteros `h_i`, que representan las alturas de cada columna de la muralla.
- `Q`: que representa la cantidad de ataques planeados por Trusia.
- `ataques`: Un arreglo de Q tuplas, cada tupla contiene tres enteros `l_i`, `r_i` y `w_i`, que representan el rango del ataque `Q_i` y el número de columnas consecutivas que se intentarán derribar en ese rango.

Retorna

- Un arreglo de Q numeros, donde cada Q_i represente la altura máxima H del cuadrilátero que Nathalie puede reforzar.

Límites

- `1 <= N, Q <= 1e5`
- `1 <= h_i <= 1e9`
- `1 <= l_i <= r_i <= N`
- `1 <= w_i <= r_i - l_i + 1`

Subtareas

- `h_i >= h_{i-1}` (12 pts)
- `N, Q <= 1000` (17 pts)
- `h_i = 1 o 2` para todo `i` (22 pts)
- `N, Q <= 5000` (19 pts)
- Sin restricciones (31 pts)

Ejemplo

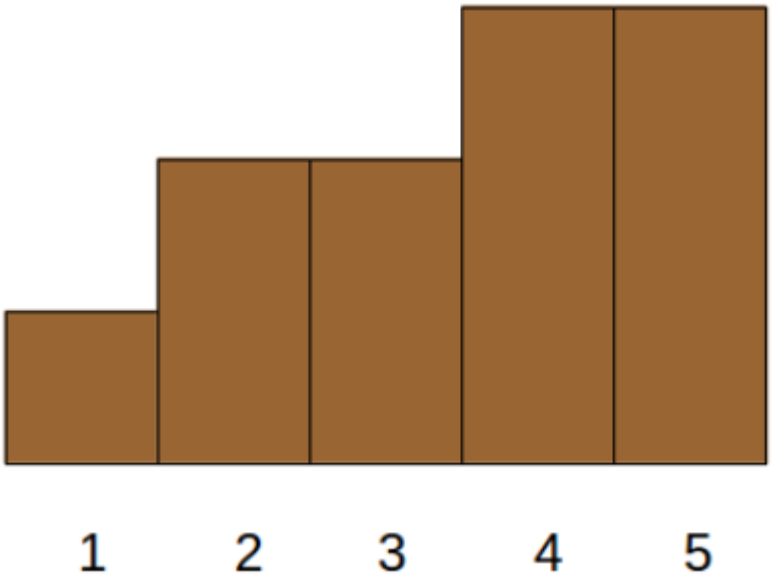
Ejemplo 1

En la siguiente llamada:

```
altura_maxima(
  5, [1, 2, 2, 3, 3], 3, [[2, 5, 3], [2, 5, 2], [1, 5, 5]]
)
```

Debe retornar `[2, 3, 1]`

la muralla se ve como en la siguiente imagen



Para la primera query, la respuesta es 2 dado que la altura maxima de muralla que puedo cubrir con un cuadrilatero de base 3, en el rango de 2 a 5, como se ve en la siguiente imagen.

