Máximo en Ventana W37576_es

Dada una secuencia de N enteros y un tamaño de ventana W, para cada posición i de la secuencia hay que encontrar el máximo de la ventana que termina en i. La ventana contiene los elementos desde la posición $\max(0, i - W + 1)$ hasta i (ambos incluidos).

Para resolver este problema de manera eficiente, usaremos una cola de prioridad (en nuestro caso, un *max-heap*) que almacena pares (valor, índice). Definimos el struct:

```
struct Elem {
    int valor;
    int index;
};
```

con la comparación:

```
bool operator>(const Elem& a, const Elem& b) {
    return a.valor > b.valor;
}
```

La técnica para poder hacer esto es la siguiente. Para cada nuevo elemento de la secuencia, lo insertamos en la cola de prioridad. Después, mientras el elemento de la cima tenga un índice fuera de la ventana (es decir, `index <= i - W`), lo extraemos. El máximo actual de la ventana será, entonces, el valor de la cima de la cola de prioridad.

Esta técnica se llama "borrado perezoso" (*lazy deletion*): los elementos que ya no pertenecen a la ventana no se eliminan inmediatamente, sino que se extraen solo cuando llegan a la cima del heap. Esto hace que el algoritmo sea muy eficiente en la práctica.

Entrada

La primera línea contiene un entero W ($W \geq 1$), el tamaño de la ventana. A continuación, una secuencia de enteros separados por espacios hasta el final de la entrada.

Salida

Para cada posición de la secuencia (empezando por la primera), el máximo de la ventana correspondiente, uno por línea.

Observación

Este es un problema de eficiencia. Una solución ingenua que recorra toda la ventana para cada posición no pasará los casos de prueba grandes, pero la solución con una cola de prioridad sí.

Hay que enviar un fichero `program.cc` que incluya `"heap.hh"`. En los ficheros públicos encontrarás `heap.hh` y `assert.hh`.

Ejemplo de entrada 1

```
3
1 3 2 5 4 1 7 2
```

Ejemplo de salida 1

```
1
3
3
5
5
5
7
7
```

Ejemplo de entrada 2

```
1
5 3 8 1
```

Ejemplo de salida 2

```
5
3
8
1
```

Información del problema

Autoría: Pau Fernández
Traducción: Pau Fernández

Generación: 2026-03-05T11:27:39.227Z