Problem Solving

Problemas y técnicas interesantes

3Sum

Dado un arreglo a de tamaño n y un entero s, la tarea es verificar si existe un trío de números en el arreglo cuya suma sea igual al valor objetivo dado.

Entrada	Salida
`6 24` `12 3 4 1 6 9`	YES
`5 50` `1 2 3 4 5`	NO
`5 0` `0 -1 2 -3 1`	YES

¿Cómo resolverlo? — Two Pointers

- Ordenar el arreglo original.
- Iterar por cada i y fijamos $a[i]$ como el primero número.
- Utilizamos 2 punteros $left = i + 1$ y $right = n - 1$ y mientras $left - right$:
  - Calcular la suma actual: $a[i] + a[left] + a[right]$
  - Si la suma es igual a s, hemos encontrado un trío.
  - Si la suma es menor que s, mover el puntero izquierdo ($left + +$) para aumentar la suma.
  - Si la suma es mayor que s, mover el puntero derecho ($right - -$) para reducir la suma.
- Si al final no se encuentra ningún trío, entonces no existe combinación válida.

¿Qué es Two Pointers?

- Técnica utilizada sobre **arreglos ordenados** para recorrer eficientemente combinaciones de elementos.
- Consiste en usar **dos índices** ($left, right$) que se mueven desde extremos opuestos.
- Permite resolver problemas como:
  - Encontrar pares o tríos con suma específica.
  - Verificar si un arreglo es palíndromo.

Vasya y la cadena

Vasya recibió una cadena de longitud n como regalo de cumpleaños. Esta cadena está compuesta de solo letras `'a'` y `'b'`

Vasya define la **belleza** de una cadena como la **longitud máxima de una subcadena (consecutiva)** que contiene **solo letras iguales**.

Puede cambiar **como máximo k caracteres** de la cadena original (de `'a'` a `'b'` o viceversa). ¿Cuál es la **máxima belleza** que puede lograr Vasya?

Link: https://codeforces.com/contest/676/problem/C

Ejemplos:

Entrada	Salida
`4 2` `abba`	4
`8 1` `aabaabaa`	5

2^Sort

Link : https://codeforces.com/problemset/problem/1692/G

Cómo resolver

1. Crear un arreglo auxiliar `b[i]` donde:
   - `b[i] = 1` si `a[i] < 2 * a[i+1]`
   - `b[i] = 0` en caso contrario
2. Queremos contar cuántos subarreglos de tamaño `k` en `b[]` tienen **suma total = k**
   → Esto indica `k` comparaciones válidas seguidas.
3. Aplicar **Sliding Window** sobre `b[]` con ventana de tamaño `k`.

Sliding Window

1. Inicializa una ventana que cubre los primeros `k` elementos.
2. Recorre la secuencia:
    - **Agrega** el nuevo elemento que entra en la ventana.
    - **Quita** el elemento que ya no está en la ventana.
3. Actualiza la respuesta en cada paso según el estado de la ventana.

Equilibrium

Link: https://codeforces.com/group/e97Gs5ZI1K/contest/620468/problem/E

Reverse Engineering

Link: https://codeforces.com/group/9CNwiex6Ir/contest/606592/problem/E

I'm Still Celebrating!

Link: https://codeforces.com/group/9CNwiex6Ir/contest/606592/problem/A

Hay muchas personas!

https://codeforces.com/group/9CNwiex6Ir/contest/530284/attachments/download/26584/mfp24_es.pdf

Find Amir!

https://codeforces.com/group/3Zw9kC8et8/contest/621025/problem/F

New Energy Vehicle

https://codeforces.com/group/3Zw9kC8et8/contest/623185/problem/L

Case of Fugitive

Link: https://codeforces.com/group/3Zw9kC8et8/contest/620967/problem/B