

CP Scripts

Estoy intentando aprender a hacer scripts en bash al mismo tiempo que intento utilizarlos para acelerar mi experiencia en la programación competitiva. Por ahora, solo he creado un script llamado `cf_run`, que ejecuta casos de prueba de ejemplo tomados de la página del problema en Codeforces y los ejecuta con el archivo de salida de tu programa, mostrando cuántos de los casos de prueba de ejemplo fueron superados.

Soy un principiante en bash scripting, así que aparentemente no pude capturar todos los errores y también el código puede contener algunos fragmentos de código innecesarios.

Scripts

- `cf_run`: Ejecuta los casos de prueba de ejemplo y muestra el estado de la salida para concursos de Codeforces.

Uso

Los scripts se han probado en Ubuntu 20.04 (si el script funciona en tu dispositivo también, siéntete libre de agregarlo aquí). Puedes seguir estos pasos para que funcionen.

1. Clona este repositorio usando el comando `git clone`.
2. Asegúrate de que todos los scripts sean ejecutables; si no lo son, puedes hacerlos ejecutables usando el comando `chmod +x nombre-del-script`.
3. Agrega este directorio a tu variable de entorno `PATH`, para poder usarlo desde cualquier lugar.
4. Sigue los pasos respectivos para cada directorio.

`cf_run`

Este script ejecutará tu programa contra los casos de prueba de ejemplo presentes en la página del problema en Codeforces, de ahí el nombre `cf_run`. Para que funcione, debes asegurarte de lo siguiente:

1. Debes crear un directorio para el concurso en el que estás participando. Digamos que estás participando en el concurso 1619, entonces debes crear un directorio con el mismo nombre usando el comando `mkdir 1619 && cd 1619`.
2. Supongamos que, para la declaración del problema A, creas un archivo llamado `a.cpp` y lo compilas. Digamos que el archivo compilado tiene el nombre `a.out`.

3. Luego, para ejecutar tu archivo, usarás el comando `cf_run a.out A`, que básicamente ejecuta casos de prueba de ejemplo para la declaración del problema A contra el archivo `a.out`.

Funcionamiento: Toma el ID del concurso del nombre de tu directorio (en este caso, 1619), luego, utilizando este ID y el segundo argumento del comando (es decir, el código del problema, aquí A), realiza un raspado de la página web <https://codeforces.com/contest/1619/problem/A>.

1 section name

D. Co-crecimiento de Secuencias

Límite de tiempo por prueba: 2 segundos

Límite de memoria por prueba: 512 megabytes

Una secuencia de enteros no negativos a_1, a_2, \dots, a_n se llama creciente si, para todo i desde 1 hasta $n-1$, todos los unos (en representación binaria) en a_i están en las mismas posiciones que los unos en a_{i+1} (en otras palabras, $a_i \& a_{i+1} = a_i$, donde $\&$ denota la operación AND a nivel de bits). Si $n = 1$, entonces la secuencia también se considera creciente.

Por ejemplo, las siguientes cuatro secuencias son crecientes:

- $[2, 3, 15, 175]$ — en binario es $[10_2, 11_2, 1111_2, 10101111_2]$;
- $[5]$ — en binario es $[101_2]$;
- $[1, 3, 7, 15]$ — en binario es $[1_2, 11_2, 111_2, 1111_2]$;
- $[0, 0, 0]$ — en binario es $[0_2, 0_2, 0_2]$.

Las siguientes tres secuencias no son crecientes:

- $[3, 4, 5]$ — en binario es $[11_2, 100_2, 101_2]$;
- $[5, 4, 3]$ — en binario es $[101_2, 100_2, 011_2]$;
- $[1, 2, 4, 8]$ — en binario es $[0001_2, 0010_2, 0100_2, 1000_2]$.

Consideremos dos secuencias de enteros no negativos x_1, x_2, \dots, x_n y y_1, y_2, \dots, y_n . Llamemos a este par de secuencias co-creciente si la secuencia $x_1 \oplus y_1, x_2 \oplus y_2, \dots, x_n \oplus y_n$ es creciente, donde \oplus denota la operación XOR a nivel de bits.

Se te proporciona una secuencia de enteros x_1, x_2, \dots, x_n . Encuentra la secuencia y_1, y_2, \dots, y_n lexicográficamente mínima tal que las secuencias x_i e y_i son co-crecientes.

La secuencia a_1, a_2, \dots, a_n es lexicográficamente menor que la secuencia b_1, b_2, \dots, b_n si existe $1 \leq k \leq n$ tal que $a_i = b_i$ para cualquier $1 \leq i < k$, pero $a_k < b_k$.

Entrada

La primera línea contiene un entero t ($1 \leq t \leq 10^4$). Luego siguen t casos de prueba.

La primera línea de cada caso de prueba contiene un entero n ($1 \leq n \leq 2 \cdot 10^5$) — longitud de la secuencia x_i .

La segunda línea contiene n enteros x_1, x_2, \dots, x_n ($0 \leq x_i < 2^{30}$) — elementos de la secuencia x_i . Se garantiza que la suma de n en todos los casos de prueba no supera $2 \cdot 10^5$.

Salida

Para cada caso de prueba, imprime n enteros y_1, y_2, \dots, y_n ($0 \leq y_i < 2^{30}$) — secuencia lexicográficamente mínima tal que es co-creciente con la secuencia dada x_i .