

Tarea 2

Profesor: Diego Arroyuelo

Ayudantes: Gabriel Carmona, Martín Crisóstomo, Nicolás Rodríguez

`gabriel.carmonat@sansano.usm.cl,`
`martin.crisostomo@sansano.usm.cl,`
`nicolas.rodriguezh@sansano.usm.cl.`

Fecha de Entrega: 15 de diciembre, 2020

Plazo máximo de entrega: 3 días.

Reglas del Juego

La presente tarea debe hacerse en grupos de 3 personas. Toda excepción a esta regla debe ser conversada con los ayudantes **ANTES** de comenzar la tarea. No se permiten de ninguna manera grupos de más de 3 personas. Pueden usarse los lenguajes de programación C, C++, Python, y Java. Esta tarea está pensada para una semana de trabajo, considerando que hay una semana intermedia de vacaciones.

Problema 1: Posicionar trampas contra un superhéroe

Un villano se encuentra preocupado por una pelea que tendrá con el superhéroe de la ciudad, por lo que piensa sabotearlo colocándole trampas. Él tiene acceso a una cantidad fija de trampas, digamos s , y quiere saber cómo posicionarlas para tener un mejor efecto. El villano ha determinado n posiciones en donde puede colocar las trampas sin ser descubierto. Dichas posiciones están representadas por una secuencia ordenada de enteros. La distancia entre dos posiciones cualquiera es la diferencia entre los valores enteros correspondientes. Se ha determinado que la mejor manera de colocar las S trampas es aquella cuya menor diferencia entre trampas sea máxima.

Por ejemplo, si las posibles posiciones para las trampas son $(1, 2, 4, 8, 9)$, y se tienen $s = 3$ trampas, entonces las posibles maneras de colocarlas (y la correspondiente diferencia mínima en cada caso) son:

- 1, 2, 4: diferencia mínima 1.
- 1, 2, 8: diferencia mínima 1.
- 1, 2, 9: diferencia mínima 1.
- 1, 4, 8: diferencia mínima 3.
- 1, 4, 9: diferencia mínima 3.
- 1, 8, 9: diferencia mínima 1.
- 2, 4, 8: diferencia mínima 2.
- 2, 4, 9: diferencia mínima 2.
- 2, 8, 9: diferencia mínima 1.

- 4, 8, 9: diferencia mínima 1.

La máxima mínima diferencia para este ejemplo es 3, correspondiente a las secuencias 1, 4, 8 y 1, 4, 9.

Diseñe e implemente un algoritmo eficiente usando la técnica de decrecer y conquistar. El tiempo de ejecución de su algoritmo debe ser $O(s \lg n)$

Hint: piense en búsqueda binaria.

Formato de Entrada

Los datos serán leídos desde la entrada standard, con el siguiente formato. La primera línea contiene dos enteros, n y s , siendo la cantidad de posiciones y cantidad de trampas a colocar, respectivamente, ($1 \leq s \leq n \leq 10^6$). Luego, la siguiente línea contiene n valores enteros, x_1, x_2, \dots, x_n , separados entre sí por un único espacio. Esos corresponden a las posiciones donde se pueden colocar las trampas, ($1 \leq x_i \leq 10^6$).

Un ejemplo de entrada es la siguiente:

```
5 3
1 4 2 9 8
```

Formato de Salida

La salida del programa debe mostrarse a través de la salida standard. La misma consiste de un único entero que indica la máxima distancia mínima entre dos trampas.

La salida correspondiente a la entrada mostrada anteriormente es la siguiente:

```
3
```

Problema 2: Calculando el Desequilibrio Total en un Arreglo

Dado un arreglo $A[1..n]$ de n valores enteros positivos, el valor de desequilibrio de A corresponde a la diferencia entre el valor máximo y mínimo almacenado en el arreglo. Su tarea es diseñar e implementar un algoritmo del tipo dividir y conquistar que permita calcular la suma de los desequilibrios de todos los subarreglos de un arreglo dado. Por ejemplo, dado el arreglo $A[1..5] = (6, 5, 2, 9, 3)$, la suma de los desequilibrios de sus subarreglos es 56, obtenida a partir de lo siguiente:

- $A[1..1] = (6)$, el valor de desequilibrio es $6 - 6 = 0$;
- $A[1..2] = (6, 5)$, el valor de desequilibrio es $6 - 5 = 1$;
- $A[1..3] = (6, 5, 2)$, el valor de desequilibrio es $6 - 2 = 4$;
- $A[1..4] = (6, 5, 2, 9)$, el valor de desequilibrio es $9 - 2 = 7$;
- $A[1..5] = (6, 5, 2, 9, 3)$, el valor de desequilibrio es $9 - 2 = 7$;
- $A[2..2] = (5)$, el valor de desequilibrio es $5 - 5 = 0$;
- $A[2..3] = (5, 2)$, el valor de desequilibrio es $5 - 2 = 3$;
- $A[2..4] = (5, 2, 9)$, el valor de desequilibrio es $9 - 2 = 7$;
- $A[2..5] = (5, 2, 9, 3)$, el valor de desequilibrio es $9 - 2 = 7$;
- $A[3..3] = (2)$, el valor de desequilibrio es $2 - 2 = 0$;
- $A[3..4] = (2, 9)$, el valor de desequilibrio es $9 - 2 = 7$;
- $A[3..5] = (2, 9, 3)$, el valor de desequilibrio es $9 - 2 = 7$;

- $A[4..4] = (9)$, el valor de desequilibrio es $9 - 9 = 0$;
- $A[4..5] = (9, 3)$, el valor de desequilibrio es $9 - 3 = 6$;
- $A[5..5] = (3)$, el valor de desequilibrio es $3 - 3 = 0$.

Su algoritmo debe tener tiempo de ejecución:

$$T(n) = \begin{cases} 2T(n/2) + \Theta(n), & n > 1; \\ 1; & n = 1. \end{cases}$$

Formato de Entrada

Los datos serán leídos desde la entrada standard, con el siguiente formato. La primera línea de la entrada contiene un entero n ($1 \leq n \leq 10^6$), que corresponde al tamaño del arreglo. Luego, la segunda línea contiene n enteros a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$), separados por un único espacio entre sí, correspondiente a los elementos del arreglo.

Un ejemplo de entrada es el siguiente:

```
6
6 5 2 9 2 2
```

Formato de Salida

La salida será a través de la salida standard. Se debe mostrar un único entero que corresponde a la suma de los desequilibrios de todos los subarreglos del arreglo dado como entrada.

La salida correspondiente al ejemplo mostrado anteriormente es la siguiente:

```
245
```

Entrega de la Tarea

La entrega de la tarea debe realizarse enviando un archivo comprimido llamado

`tarea2-apellido1-apellido2-apellido3.tar.gz`

(reemplazando sus apellidos según corresponda), o alternatively usando formato zip, en el sitio Aula USM del curso, a más tardar el día 15 de diciembre, 2020, a las 23:59:00 hrs (Chile Continental), el cual contenga:

- Los archivos con el código fuente necesarios para el funcionamiento de la tarea.
- `NOMBRES.txt`, nombre y ROL de cada integrante del grupo.
- `README.txt`, instrucciones de compilación (en caso de ser necesarias).
- `Makefile`, archivo de compilación automática (en caso de ser necesario).

Rúbrica de Evaluación

Para corregir la tarea se evaluarán los aspectos y descuentos estipulados en la siguiente tabla. Cada ítem tendrá su puntaje del 1 al 100 y será ponderado de acuerdo a los porcentajes indicados. Los descuentos son aplicados sobre la nota final.

FORMATO I/O	20%
Uso correcto de la entrada y salida estándar	50%
Lectura e impresión en el formato solicitado	50%
TÉCNICA DE PROGRAMACIÓN	80%
Uso de la estrategia de programación solicitada en el enunciado	40%
Eficiencia del algoritmo implementado	60%
ÍTEM	DESCUENTO
No entregar archivos en <code>.tar.gz</code> o <code>.zip</code>	-15
Código fuente está desordenado	-15
No incluir los archivos de documentación solicitados	-15
Correcta administración y manejo de memoria según lenguaje elegido	-30
Copiar (desde algún sitio web o compañero)	-100
Tarea no compila	-100
Tarea no termina su ejecución correctamente	-100
Atraso	-10 por día
<i>Warnings</i> de compilación	-5 por <i>warning</i>