

CS2023 - Aula de Ejercicios N° 5
Brenner H. Ojeda Rios
Semestre 2024-1

Se sugiere que cada estudiante trate de resolver los ejercicios de forma **individual** y luego los discuta en grupo.

Para la solución de estos ejercicios **no** debe usar el contenedor `map` de STL. **Usted, debe crear su propia tabla hash.**

Ejercicios

1. (5 pts) Dada una cadena de texto, desea utilizar los caracteres del texto para formar tantas instancias de la palabra “balloon” como sea posible. Puedes utilizar cada carácter del texto como máximo una vez. Devuelve el número máximo de instancias que se pueden formar. Vea el siguiente ejemplo:

- Ejemplo 1:

nlaebolko

Input: text = "nlaebolko"

Output: 1

- Ejemplo 2:

loonbalxballpoon

Input: text = "loonbalxballpoon"

Output: 2

Restricciones:

- $1 \leq \text{text.longitud} \leq 10^4$
 - El texto consta únicamente de letras minúsculas en inglés.
2. (6 pts) Se le proporcionan las cadenas *clave* y *mensaje*, que representan una clave de cifrado y un mensaje secreto, respectivamente. Los pasos para decodificar el mensaje son los siguientes:
 - Utilice la primera aparición de las 26 letras minúsculas inglesas en *clave* como orden de la tabla de sustitución.
 - Alinee la tabla de sustitución con el alfabeto inglés.
 - Luego, cada letra del *mensaje* se sustituye utilizando la tabla.
 - Los espacios “ ” se transforman en sí mismos.

Por ejemplo, dada la clave = “happy boy” (la clave real tendría al menos una instancia de cada letra en el alfabeto), tenemos la tabla de sustitución parcial de ($'h' \rightarrow 'a'$, $'a' \rightarrow 'b'$, $'p' \rightarrow 'c'$, $'y' \rightarrow 'd'$, $'b' \rightarrow 'e'$, $'o' \rightarrow 'f'$). Su tarea es devolver el *mensaje* decodificado. Vea los siguientes ejemplos para una mejor comprensión.

- Ejemplo 1

Input: clave = “the quick brown fox jumps over the lazy dog”, mensaje = “vkbs bs t suepuv”

Output: “this is a secret”

t	h	e	q	u	i	c	k	b	r	o	w	n	f	x	j	m	p	s	v	l	a	z	y	d	g
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z

Figura 1: Tabla de sustitución asociada al ejemplo 1.

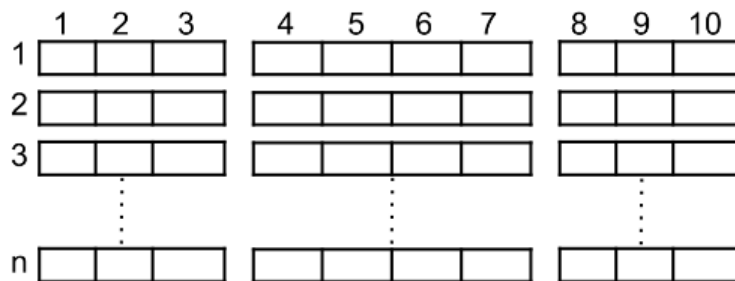
■ Ejemplo 2

Input: clave = “abcdefghijklmnopqrstuvwxyz”, mensaje = “i love lourdes”

Output: “i love lourdes”

Restricciones:

- $26 \leq \text{clave.longitud} \leq 2000$
 - La clave consta de letras minúsculas en inglés y ' '.
 - La clave contiene todas las letras del alfabeto inglés (de la 'a' a la 'z') al menos una vez.
 - $1 \leq \text{mensaje.longitud} \leq 2000$.
 - El mensaje consta de letras minúsculas en inglés y ' '.
3. (9 pts) Un cine tiene n filas de asientos, numeradas del 1 al n y hay diez asientos en cada fila, etiquetados del 1 al 10 como se muestra en la figura.

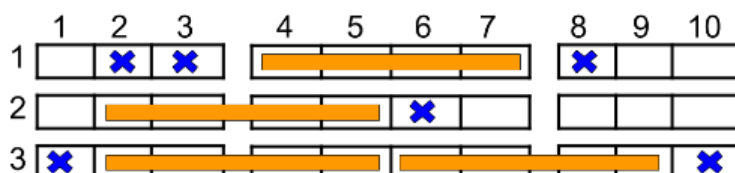


Dada la matriz de asientos reservados que contiene el número de asientos ya reservados, por ejemplo, asientos reservados $[i] = [3,8]$ significa que el asiento ubicado en la fila 3 y etiquetado con 8 ya está reservado.

Devuelve el número máximo de grupos de cuatro personas que puedes asignar en las butacas del cine. Un grupo de cuatro personas ocupa cuatro asientos contiguos en una sola fila. Los asientos al otro lado de un pasillo (como $[3,3]$ y $[3,4]$) no se consideran adyacentes, pero existe un caso excepcional en el que un pasillo divide a un grupo de cuatro personas; en ese caso, el pasillo divide un grupo de cuatro personas en el medio, lo que significa tener dos personas a cada lado. Vea los siguientes ejemplos para una mejor comprensión.

■ Ejemplo 1

Input: $n = 3$, reservedSeats = $[[1,2],[1,3],[1,8],[2,6],[3,1],[3,10]]$



Output: 4

Explicación: La figura anterior muestra la asignación óptima para cuatro grupos, donde los asientos marcados con azul ya están reservados y los asientos contiguos marcados con naranja son para un grupo.

■ **Ejemplo 2**

Input: $n = 2$, reservedSeats = [[2,1],[1,8],[2,6]]

Output: 2

■ **Ejemplo 3**

Input: $n = 4$, reservedSeats = [[4,3],[1,4],[4,6],[1,7]]

Output: 4

Restricciones:

- $1 \leq n \leq 10^9$
- $1 \leq \text{reservedSeats.length} \leq \min(10 \times n, 10^4)$
- $\text{reservedSeats}[i].\text{length} = 2$
- $1 \leq \text{reservedSeats}[i][0] \leq n$
- $1 \leq \text{reservedSeats}[i][1] \leq 10$
- Todos los reservedSeats[i] son distintos.