

## S2-Simulacro-Entrevista-2

### \*SiE2\* - Qué rico un taco - Sencillo

#### \*SiE2\* - Qué rico un taco - Sencillo

Ficheros requeridos: [taco.py](#) (Descargar)

Tipo de trabajo: Individual

Eres el dueño de un puesto de tacos. Sin embargo, en estas últimas semanas tus tacos se han vuelto realmente famosos dentro de la ciudad gracias a su sabor único e inigualable. Es por ello que todos los días se forman largas filas de espera. Debido a esto, te gustaría llevar un control del negocio. Para hacer la tarea más fácil, has decidido crear un programa que te ayude en esta tarea.

#### Entrada

En la primera línea recibirás un entero N, el número de operaciones a realizar. En las siguientes N líneas recibirás un entero representando la operación a realizar:

- Si el entero es un 1, significa que un nuevo cliente se forma en la fila, por lo que deberás leer un entero t representando el número de tacos que desea ordenar.
- Si el entero es un 2, significa que el cliente que se encuentra al frente de la fila será atendido.
- Si el entero es un 3, significa que quieres saber el número de clientes que se encuentran formados actualmente en la fila.
- Si el entero es un 4, significa que quieres saber el número de tacos que has vendido hasta ahora.

#### Salida

Por cada una de las entradas del tipo 3 y 4, deberás imprimir un entero por línea representando la respuesta a la pregunta correspondiente.

#### Ejemplo1

##### Entrada

```
9
1 4
1 3
1 10
3
1 5
2
2
1 9
4
```

##### Salida

```
3
7
```

### \*SiE2\* - Máxima distancia - Sencillo

## \*SiE2\* - Máxima distancia - Sencillo

Ficheros requeridos: [distancia.py](#) (Descargar)

Tipo de trabajo: Individual

Dada una cantidad  $n$  de números repetidos, la tarea es encontrar la máxima distancia entre dos ocurrencias de un número.

**Entrada**

Un número  $n$ ,  $1 \leq n \leq 1e^5$ , la siguiente línea los  $n$  números

**Salida**

Un número entero indicando cuál es la máxima distancias entre dos ocurrencias de un número

*Ejemplo:*

**Entrada:**

12

3 2 1 2 1 4 5 8 6 7 4 2

**Salida:**

10

En este caso solo tenemos 3 números que se repiten, si analizamos cuál es la máxima distancia notamos que:

Máxima distancia para 2 es  $11-1 = 10$

Máxima distancia para 1 es  $4-2 = 2$

Máxima distancia para 4 es  $10-5 = 5$

## \*SiE2\* - Árboles binarios sin balancear - Sencillo

### \*SiE2\* - Árboles binarios sin balancear - Sencillo

Ficheros requeridos: [arbol.py](#) (Descargar)

Tipo de trabajo: Individual

A diferencia de los árboles binarios búsqueda balanceados, los árboles binarios de búsqueda no balanceados son muy fáciles de programar: cuando insertas un nuevo elemento, vas bajando por el árbol (hacia el subárbol izquierdo si el elemento a insertar es menor que el del nodo actual y hacia el subárbol derecho en caso contrario) hasta colocar el elemento en el primer lugar que esté libre. Escribe un programa que inserte una secuencia de enteros en un árbol binario de búsqueda no balanceado (inicialmente vacío) y que responda para cada entero con cuántos elementos del árbol se comparó durante su inserción.

**Entrada**

Un entero  $N$  seguido de una secuencia de  $N$  enteros distintos, cada uno en el rango de 1 a  $N$

**Salida**

Para cada uno de los  $N$  enteros, un entero que sea el número de comparaciones con elementos del árbol que se necesitaron para insertar el entero.

*Ejemplo1*

**Entrada**

7

1 2 3 4 5 6 7

**Salida**

0

1

2

3

4

5

6

## \*SiE2\* - Hallar el número de islas - Sencillo

## \*SiE2\* - Hallar el número de islas - Sencillo

Ficheros requeridos: numIslas.py (Descargar)

Tipo de trabajo: Individual

### HALLAR EL NÚMERO DE ISLAS EN UN GRAFO

Dada una matriz 2D binaria, calcule el número de islas y retorne ese número.

*¿Qué es una isla?*

Un grupo de unos conectados forma una isla.

Por ejemplo, la siguiente matriz tiene 5 islas:

```
{1, 1, 0, 0, 0},
{0, 1, 0, 0, 1},
{1, 0, 0, 1, 1},
{0, 0, 0, 0, 0},
{1, 0, 1, 0, 1}
```

Entrada: mat [] []

- Matriz de dos dimensiones con la información del grafo (se pasa como parámetro)

Salida: int

- Entero que se debe retornar, el cuál representa el número de islas en el grafo

## Ficheros requeridos

numIslas.py

```
1 def countIslands(mat) -> int:
2     # codigo aqui
```