

Examen Parcial

Materia: Programación Superior IMT-231

Docente: Francisco Suárez

Problema 1

Escriba una función llamada **esNúmeroFeliz** que retorne un booleano que determine si un entero dado es “feliz” o no. Un entero es “feliz” si la suma de los cuadrados de sus dígitos, repetidamente, eventualmente nos lleva a 1.

Por ejemplo, 139 es feliz porque:

- $1^2 + 3^2 + 9^2 = 91$
- $9^2 + 1^2 = 82$
- $8^2 + 2^2 = 68$
- $6^2 + 8^2 = 100$
- $1^2 + 0^2 + 0^2 = 1$

En contraste, 4 no es feliz porque:

- $4^2 = 16$
- $1^2 + 6^2 = 37$
- $3^2 + 7^2 = 58$
- $5^2 + 8^2 = 89$
- $8^2 + 9^2 = 145$
- $1^2 + 4^2 + 5^2 = 42$
- $4^2 + 2^2 = 20$
- $2^2 + 0^2 = 4$
- ...

Problema 2

Escriba una función llamada **listaAmigos** que acepte como parámetro un archivo de texto, y lea del archivo las relaciones de amistad almacenándolas en un diccionario donde el nombre de cada persona sea una llave, y el valor asociado a esa llave se un set de los amigos de esa persona. Las amistades son bidireccionales: si Pepe es amigo de Pedro, Pedro es amigo de Daniel.

El archivo contiene una relación de amistad por línea, que está formada de dos nombres. Los nombres están separados por un solo espacio. Puede asumir

que el archivo de texto existe en un formato válido. Si un archivo llamado **amigos.txt** contiene:

Pepe Pedro

Ana Pepe

Entonces una llamada a **listaAmigos("amigos.txt")** debería retornar un diccionario con el siguiente contenido:

```
{"Ana": {"Pepe"}, "Pedro": {"Pepe"}, "Pepe": {"Ana", "Pedro"}}
```

Problema 3

Escriba una función llamada **tieneInversoRepetido** que acepte dos arrays de enteros **a1** y **a2** y sus longitudes como parámetros. La función retorna **True** si **a1** contiene todos los elementos de **a2** en orden reverso al menos dos veces (y **False** en caso contrario).

Por ejemplo, si **a2** contiene los elementos {1, 2, 3} y **a1** contiene los elementos {6, 3, 2, 1, 4, 1, 3, 2, 1, 5}, llamar a **tieneInversoRepetido(a1, 10, a2, 3)** debería retornar **True**.

Asuma que ambos arrays tienen una longitud de por lo menos 1. Esto significa que el inverso más corto posible puede ser de longitud 1, es decir un solo elemento (que es su propio inverso). Una secuencia que es palíndromo (igual en orden inverso) debería ser considerado en sus cálculos. Por ejemplo, si **a1** es {6, 1, 2, 1, 4, 1, 2, 1, 5} y **a2** es {1, 2, 1}, su función debería retornar **True**. Las dos ocurrencias del inverso pueden estar solapadas, como en el 4 caso.

a1 = {6, 1, <u>2, 1</u> , 3, 1, 3, <u>2, 1</u> , 5}; a2 = {1, 2};	tieneInversoRepetido(a1, 10, a2, 2) retorna True
a3 = { <u>5, 8, 4</u> , 18, 5, 42, 4, 8, 5, 5}; a4 = {4, 8, 5};	tieneInversoRepetido(a3, 10, a4, 3) retorna False
a5 = {6, <u>3, 42</u> , 18, 12, 5, <u>3, 42</u> , <u>3, 42</u> }; a6 = {42, 3};	tieneInversoRepetido(a5, 10, a6, 2) retorna True
a7 = {6, <u>1, 2, 4, 2, 1, 2, 4, 2, 1</u> , 5}; a8 = {1, 2, 4, 2, 1};	tieneInversoRepetido(a7, 11, a8, 5) retorna True
a9 = { <u>0, 0</u> }; aa = {0};	tieneInversoRepetido(a9, 2, aa, 1) retorna True
ab = {8, 9, 2, 1}; ac = {5, 7, 1, 2, 9, 8};	tieneInversoRepetido(ab, 4, ac, 6) retorna False

Problema 4

Escriba una función llamada **embarqueBinario** que encuentre el ID de asiento mayor en un pase a bordo a un avión que usa un algoritmo específico descrito a continuación. Su función lee datos de un archivo que en cada línea contiene la especificación para un asiento en una secuencia de 10 caracteres:

```
FBFBBFFRLR
BFFFBBFRRR
BBFFBBFRLR
FFFBBBFRRR
```

Los primeros 7 caracteres describen el movimiento hacia la parte delantera o trasera del avión con respecto a filas, y los últimos 3 caracteres se refieren a los movimientos de derecha o izquierda en una fila en particular. Un ejemplo sería FBFBBFFRLR. El avión tiene 128 filas (numeradas del 0 al 127 desde el frente hasta el fondo). Cada carácter de los primeros 7 indica en cuál de las mitades está la fila del asiento; F para la primera mitad (de las filas 0 a la 63) y B para la otra mitad (filas 64 a 127). El siguiente carácter indica la siguiente mitad, y así sucesivamente hasta que queda exactamente una fila.

Por ejemplo, considere los 7 primeros caracteres del asiento FBFBBFFRLR: Empiece considerando todo el rango, filas 0 a 127. Luego:

- F significa tomar la primera mitad, manteniendo filas 0 a 63.
- B significa tomar la segunda mitad, manteniendo filas 32 a 63.
- F significa tomar la primera mitad, manteniendo filas 32 a 47.
- B significa tomar la segunda mitad, manteniendo filas 40 a 47.
- B significa tomar la segunda mitad, manteniendo filas 44 a 47.
- F significa tomar la primera mitad, manteniendo filas 44 a 45.
- B significa tomar la primera mitad, manteniendo la fila 44.

Los tres últimos caracteres serán L o R estos especifican una de las 8 columnas de asientos en el avión (enumeradas del 0 al 7). El mismo proceso se aplica, esta vez sólo con tres pasos. L significa la primera mitad, mientras R significa la segunda mitad.

Por ejemplo, considere los 3 últimos caracteres de FBFBBFFRLR. Empiece considerando todo el rango, columnas 0 a 8. Luego:

- R significa tomar la segunda mitad, manteniendo columnas 4 a 7.
- L significa tomar la primera mitad, manteniendo columnas 4 a 5.
- R significa tomar la segunda mitad, manteniendo la columna 5.

Por lo tanto, decodificar FBFBBFFRLR revela que el asiento está en la fila 44, columna 5. Para computar el ID del asiento, multiplique la fila por 8, y sume la columna. En este ejemplo, el asiento tiene un ID de $44 \cdot 8 + 5 = 357$.

El objetivo de esta función es encontrar y retornar el **ID mayor** en la data de entrada. Para la data al inicio del problema, el ID mayor viene de la línea BBFFBBFRLL, la cual corresponde a la fila 102, columna 4, con ID 820. Por lo tanto, la llamada a `embarqueBinario("embarque.txt")` debería retornar 820.

Puede asumir que el archivo existe y puede ser leído, y que sigue el formato anteriormente descrito, y que contiene al menos una línea.

Problema 5

Describa en sus propias palabras las diferencias entre las estructuras de datos Array desarrollada en clase, y list, que es nativa del lenguaje Python. Puede usar ejemplos concretos de uso de métodos.