**Taller**

Miguel Sosa, Sebastián Contreras y Sebastián Rendón

1. (10) Se desea eliminar todos los números duplicados de una lista

Por ejemplo

Si se le ingresan los valores [4,7,11,4,9,5,11,7,3,5]

Se debe cambiar a [4,7,11,9,5,3]

1. (10) Elimine los elementos duplicados de un vector ordenado.

¿Cuál es la eficiencia del método?

Compárela con la eficiencia del punto 1

**R/** La eficiencia del método 1 es de O(n2) y la del método 2 es de O(n)

1. (5) Dada la siguiente lista

[47,3,21,32.56,92]

Después de 2 “pasadas” de un algoritmo de ordenación, la lista ha quedado dispuesto así

[3,21,47,32,56,92]

¿Qué algoritmo de ordenación se está utilizando (selección, burbuja o inserción)?

Justifique su respuesta

**R/** El algoritmo que se está usando es el de burbuja, ya que, en la primera pasada, el 47 burbujea de la posición 0 a la posición 1, quedando así el arreglo [3,47,21,32,56,92] y en la segunda pasada vuelve a burbujear de la posición 1 a la posición 2 y el arreglo queda [3,21,47,32,56,92]

1. (10) Utilizar el algoritmo de ordenación SHELL, encuentre las pasadas e intercambios que se realizan para la ordenación de la siguiente lista





1. (10) Dada una lista voto [0.......n-1], donde cada elemento de lista representa un voto en las elecciones. Suponga que cada voto se da como un número entero que representa el ID del candidato elegido. Desarrolle un algoritmo para determinar quién gana la elección. Determine la complejidad del algoritmo

**R/** La complejidad que el algoritmo tiene es de O(n).

1. (10) Se cuenta con una lista de tuplas

futbolistasTup = [(1, "Casillas"), (15, "Ramos"), (3, "Pique"), (5, "Puyol"), (11, "Capdevila"), (14, "Xabi Alonso"), (16, "Busquets"), (8, "Xavi Hernández"), (18, "Pedrito"), (6, "Iniesta"), (7, "Villa")]

si se aplica futbolistasTup.sort(key=lambda futbolista: futbolista [0])

1. Que resultado se obtiene al aplicar el método .sort

**R/** Lo que hace este método es ordenar la tupla, ya sea en orden ascendente o descendente

1. Que se está especificando en los parámetros (key=lambda futbolista: futbolista [0])

**R/** Lo que se especifica con los parámetros es como se va a ordenar la tupla, es decir, si tiene futbolista [0] se va a ordenar en torno a los primeros elementos de la tupla, en este caso, los números de los jugadores

1. Aplique este método a las listas de los puntos 1,3, 4. Que conclusión puede obtener

**R/** Que el método no sirve, debido a que estas no son tuplas

1. Por favor según opinión realice una tupla con los mejores inventos del 2019.

Donde usted califica el que más le gusta o le parece importante.

**Anotación** la escala con la que usted cuenta es de 1 a 100 (no tiene que asignar ninguno de los extremos si no lo desea)

**R/** inventos = [(1, "Prótesis robótica de mano"), (2, "Zapatos reciclables"), (3, "Domicilio automatizado"), (4, "Airpods")]

1. (10) Diseñe e implemente una función para encontrar todos los valores negativos dentro de una lista dada. Tu función debería devolver una nueva lista que contiene los valores negativos.

¿Cuándo ocurre el peor de los casos y cuál es el tiempo de ejecución para ese caso?

R/ El peor caso sería cuando todos los elementos de la lista son negativos, ya que tendría que hacer más operaciones, y para la lista [-3,-1,-15,-16,-17,-15] el tiempo de ejecución fue 0.00769840000430122 segundos.

1. (5) Dada la siguiente lista de números:

[21, 1, 26, 45, 29, 28, 2, 9, 16, 49, 39, 27, 43, 34, 46, 40]

¿Cuál sería el resultado a la lista que después de 3 llamadas recursivas de ordenamiento por mezcla?

**R/** Se tendrían dos listas, cada una de dos elementos, la mitad izquierda con [21, 1] y la mitad derecha con [26, 45]

¿Por qué?

**R/** Porque cuando se llama recursivamente, lo que hace es dividir la lista en dos, y así sucesivamente hasta no poder dividir más

1. (5) Dado el siguiente código evalué su complejidad.

Determine que hace este esté algoritmo

Esplique cada uno de sus funciones y sus clases

**R/** El código cumple varias funciones, este puede crear una instancia de conjunto vacía, también te da el número de ítems en una línea, determina si un hay un elemento en x línea, te deja eliminar un elemento de una línea y por último determina si una línea es una sublinea de setB. También, devuelve un iterador para recorrer la lista de elementos y por último encuentra la posición de un elemento en la lista y su complejidad es O(n).



1. (10) Construya un algoritmo para encontrar un valor específico en una matriz de valores ordenada por filas y columna.

El algoritmo toma como entrada una matriz de valores donde cada fila y cada columna están en orden, junto con un valor para ubicar en esa matriz. Devuelve si ese elemento existe en la matriz.

Por ejemplo, dado la siguiente matriz y buscar el 7, el algoritmo daría como resultado **sí**

Pero si se pide encontrar el número 0, el algoritmo daría como resultado **no**



1. (10) se tiene una lista A con 100 elementos A [ a1……a100]

B de 60 elementos B [ b1……b60]

Se desean resolver las siguientes tareas

1. Ordenar cada lista aplicando el método Quicksort
2. Crear una lista C que sea la unión de la lista A y B
3. Ordenar la lista C y visualizarla
4. (5) Investigue en qué consisten las técnicas de ordenación Radixsort y Binsort, en lo posible haga un ejemplo

**R/** ***Radix Sort:*** Es un método de ordenamiento que procesa los dígitos de los números individualmente, este, empezando por el menos significativo (el de más a la derecha), estos números se van organizando en una lista del 0-9, según sea el numero menos significativo es su posición (es decir, si tienes el número 39, la posición de esta será la novena posición). Después de terminar esto, los números según quedaron ordenados se devuelven a como estaban, ahora, se hará el mismo proceso, pero con la diferencia de que será con el número más significativo (el de la izquierda), al final, la lista quedará ordenada.

Ejemplo: Con el siguiente orden de números: 35, 62, 56, 3 y 16.

Paso #1:

Diagrama, Esquemático

Descripción generada automáticamente

Paso #2:

Imagen que contiene Texto

Descripción generada automáticamente

Paso #3:

Diagrama

Descripción generada automáticamente con confianza baja

Paso #4:

Diagrama

Descripción generada automáticamente

***Binsort:*** el ordenamiento Binsort o también llamado Bucketsort es un algoritmo que distribuye todos los elementos a ordenar en un numero x de casilleros. Cada uno de los casilleros tiene sus restricciones, por ejemplo, si tengo números del 1 al 10, se podría usar 2 casilleros, estos con diferentes reglas, por ejemplo, en el primer casillero solo entran números del 1-5 y el otro del 6-10.