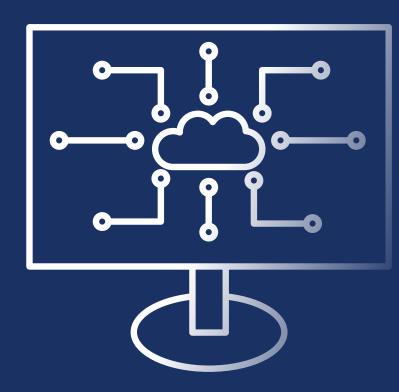
Introducción a la algoritmia



Lic. Julia Monasterio





Clase Nº8

TEMAS

- Números al azar
- Copia de listas
- Búsqueda secuencial
- Ordenamiento de vectores -Método Selección



Números al azar



Son números generados, o inventados por la computadora

Se utilizan cuando se requiere un factor de azar, por ejemplo en videojuegos, criptografía o simulación de eventos

UADE

Números al azar

 Python tiene la posibilidad de generar números al azar a través del módulo random

• ¿Que es un modulo?

Es un conjunto de funciones que ya vienen listas para usar, y que se agregan a nuestro programa a través de la instrucción **import**



Práctica en clase

Realizar en Python Escribir un programa para simular el lanzamiento de un conjunto de datos, utilizando números al azar





Solución en Python

```
import random

cuantos= int(input("¿Cuántos dados desea lanzar?\n"))

for i in range(cuantos):
    dado= random.randint(1,6)
    print(dado,end=" ")
```



Aspectos importantes

- Importación de módulos: todo módulo que desee utilizarse debe ser importado (incluido) al comienzo del programa utilizando la instrucción import
- <u>Función random.randint(minimo, maximo)</u>: genera un número entero al azar entre los límites suministrados, ambos incluidos.
 - El nombre de la función debe ir precedido por el del módulo, separados por un punto.



Aplicaciones

Los números al azar los vamos a utilizar para cargar vectores o listas, a fin de evitar el ingreso de tantos valores a través del teclado.





 Las listas no pueden copiarse directamente como si lo hacemos con las variables





Tratar de hacerlo provocará resultados inesperados

```
lista1 = [1,2,3]

lista2=lista1

lista2.append(4)

print(lista1)
print(lista2)
[1, 2, 3, 4]
[1, 2, 3, 4]
```



• El error sucede porque al copiar las listas de esa forma en realidad no se esta creando una lista a partir de otra, sino que ambas listas (**lista1 y lista2**) apuntan a la misma ubicación de memoria.

 Por este motivo cualquier cambio que se haga en una de ellas se reflejará en la otra

Entonces....¿como?





 Por ahora lo vamos a aprender con un ciclo, es decir recorreremos y asignaremos en un ciclo

```
lista1 = [1,2,3]

lista2=[]

for i in range(len(lista1)):
    lista2.append(lista1[i])

lista2.append(4)

print(lista1)
print(lista2)
[1, 2, 3]
[1, 2, 3]
```



Búsqueda secuencial



 La búsqueda secuencial es la más sencilla de las búsquedas que puedan realizarse sobre una lista

 Consiste en ir recorriendo la lista elemento por elemento hasta encontrar el valor buscado o hasta llegar al final, lo que significa que el valor no se encontraba presente



Práctica en clase

Realizar en Python

Cargar una lista con números al azar entre 1 y 100, donde la cantidad de elementos será ingresada por el usuario.

Luego se solicita leer un valor y buscarlo en la lista mediante una función, devolviendo su ubicación o -1 si no se la encuentra





Función para cargar lista

```
import random
def cargarLista(cantidad):
    lista=
    for i in range(cantidad):
        lista.append(random.randint(1,100))
    return lista
```



Función para buscar un valor en la lista

```
def busquedaSecuencial(lista,dato):
    i=0
    while i<len(lista) and lista[i]!=dato:
        i+=1
    if i< len(lista):</pre>
        return i
    else:
        return -1
```



Función para imprimir una lista

```
def imprimirLista(lista):
    for i in range(len(lista)):
        print(lista[i], end=" ")
```



Programa principal:

```
#Programa Principal
cant = int(input("¿Cuantos elementos desea cargar en la lista?\n"))
lista=cargarLista(cant)
imprimirLista(lista)
numBuscar = int(input("\nIngrese el valor que desea buscar\n"))
posicion= busquedaSecuencial(lista,numBuscar)
if posicion>=0:
    print("El elemento se encontro en la siguiente posicion: ",posicion)
else:
    print("El elemento no se encontro en la lista")
```



Ordenamiento de vectores

UADE

- El ordenamiento o clasificación de elementos ha sido una necesidad desde hace mucho antes de la aparición de la informática
- La naturaleza monótona y repetitiva de esta tarea la vuelve ideal para ser realizada mediante computadoras
- Los elementos que desea ordenar deben estar almacenados en una estructura de datos, y por eso se suelen usar arreglos (listas o vectores)

UADE

¿Como lo hacemos?

Existen muchos métodos de ordenamiento.

En esta materia visualizaremos tres:

- Selección
- Intercambio
- Inserción

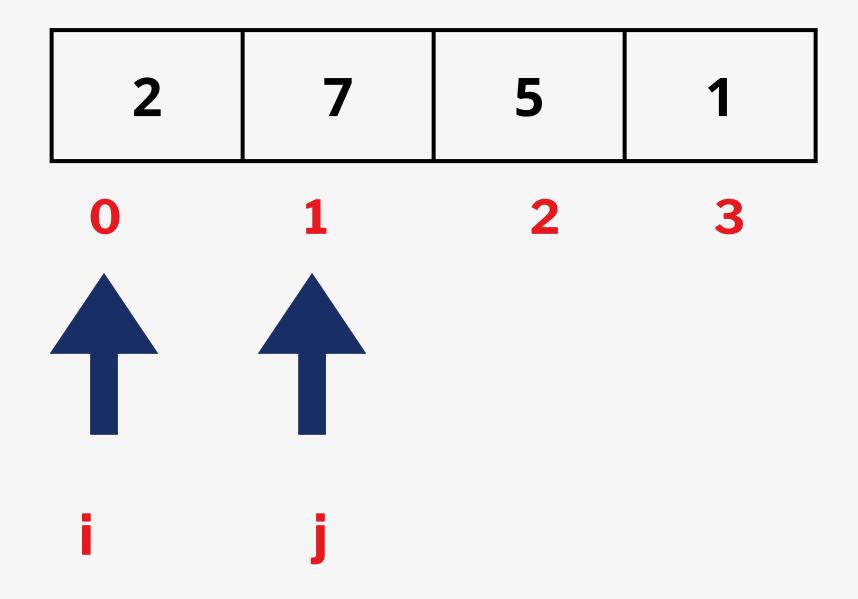


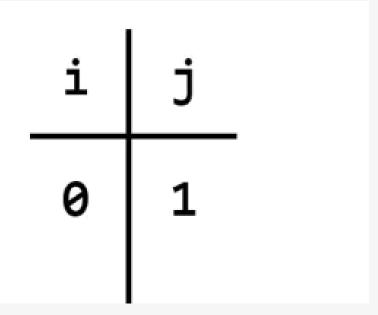
Método de selección

• Consiste en buscar el menor elemento de todo el arreglo e intercambiarlo con el de la primera posición

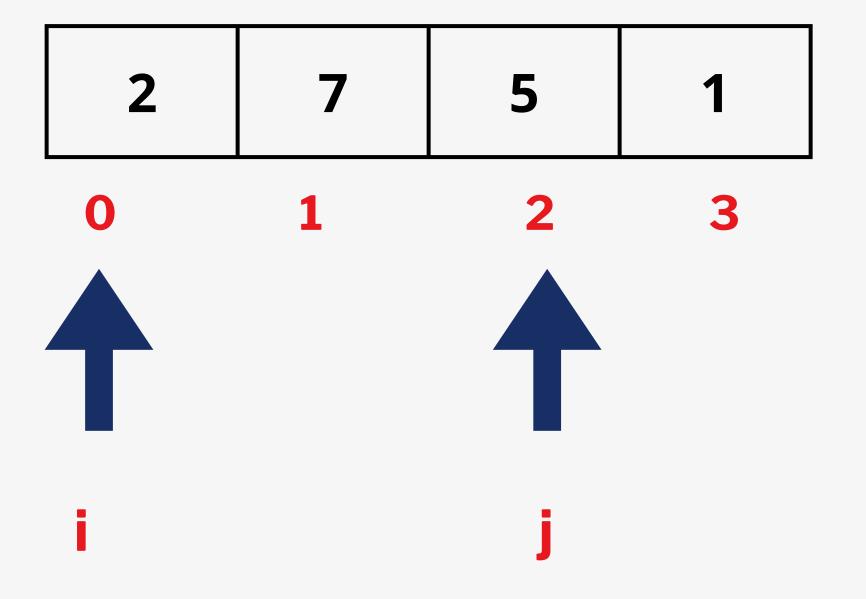
 Luego se busca el segundo menor elemento y se lo intercambia con el de la segunda posición, y así sucesivamente

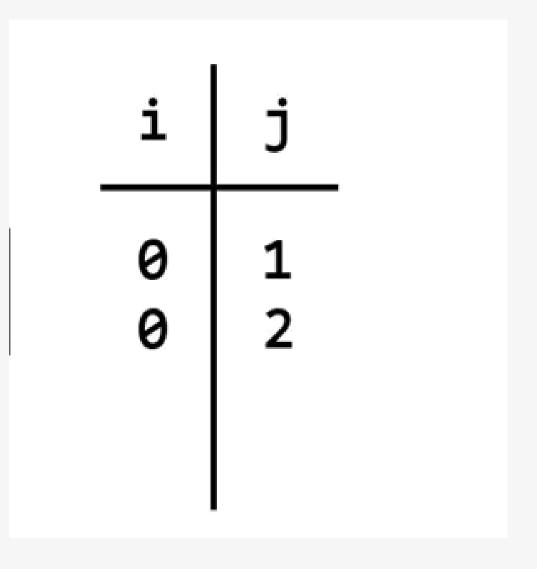






UADE



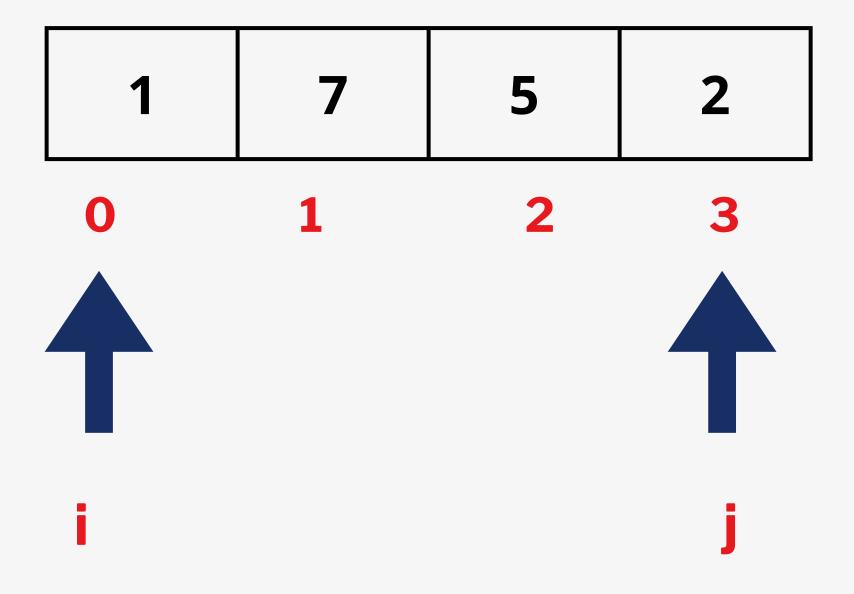


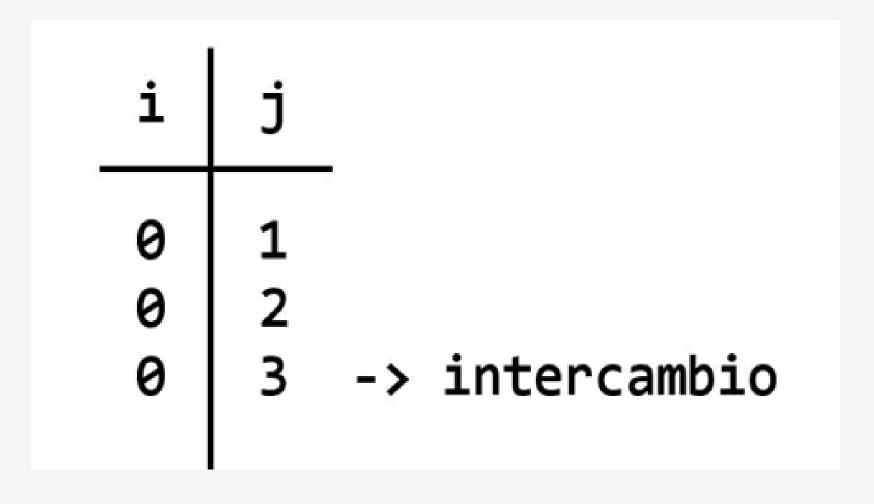


2	7	5	1
0	1	2	3

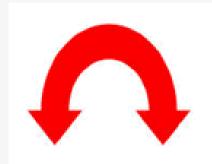
i	j
0 0 0	1 2 3 -> intercambio

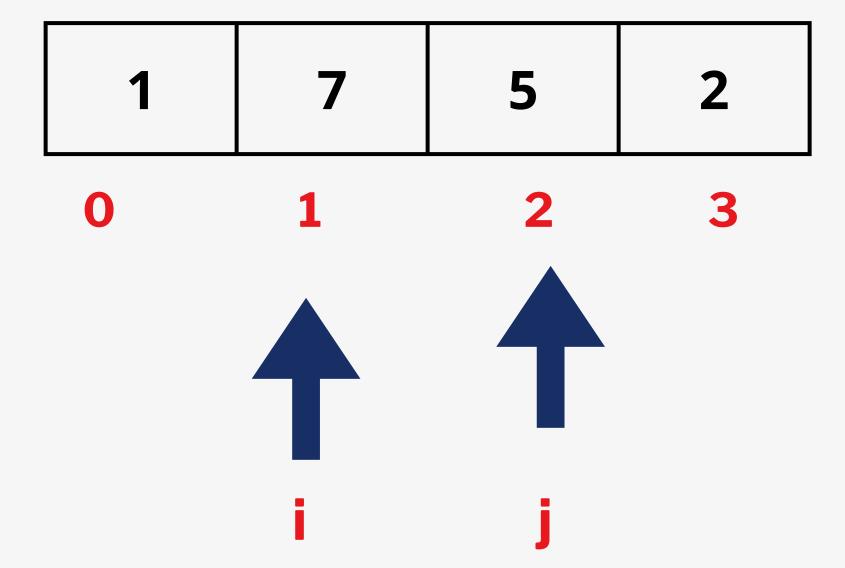












i	j	
1	2	-> intercambio



Metodo de ordenamiento

```
def metodoSeleccion(lista):
    largo=len(lista)
    for i in range(largo-1):
        for j in range(i+1, largo):
            if lista[i]>lista[j]:
                aux=lista[i]
                lista[i]=lista[j]
                lista[j]=aux
```

Lic. Julia Monasterio

UADE

Resúmen de la clase

- Números al azar
- Copia de listas
- Búsqueda secuencial
- Ordenamiento de vectores Método de selección





EJERCITACIÓN

Objetivos

- Introducir el concepto de estructuras de datos
- Familiarizarse con el uso de listas en Python, conocidas como arreglos o vectores en otros lenguajes de programación





Ejercitación

• Realizar ejercicios 8,9,10 y 11





Ejercitación

- **Ejercicio 3:** Determinar si la lista es capicúa (palíndromo). Una lista capicúa se lee de igual modo de izquierda a derecha y de derecha a izquierda. Por ejemplo, [2, 7, 7, 2] es capicúa, mientras que [2, 7, 5, 2] no lo es.
- **Ejercicio 4:** Escribir una función para contar cuántas veces aparece un valor dentro de la lista. La función recibe como parámetros la lista y el valor a buscar, y devuelve un número entero.
- <u>Ejercicio 5:</u> Desarrollar una función que reciba la lista como parámetro y devuelva una nueva lista con los mismos elementos de la primera, pero en orden inverso. Por ejemplo, si la función recibe [5, 7, 1] debe devolver [1, 7, 5]



Muchas gracias!

Consultas?

