Programación

Clase 09

U4

Arreglos
Algoritmos con listas

Máximo en una lista

¿Cómo calculamos el máximo normalmente?

¿Y como sería una función que hiciera eso en una lista?

```
def maximo(lista):
    max_ = -1
    for i in lista:
        if i > max_:
            max_ = i
    return max_

milista = [20,20,50,70,50,30,50,42,42,50]
print(maximo(milista))
```

Los dos máximos de una lista

¿Qué pasa si queremos saber cuales son los dos valores máximos de una lista?

¿Cómo lo podríamos hacer?

Encontrando el mayor y luego encontrando un valor que sea el segundo mayor comparándolo con el mayor ya encontrado.

¿Qué pasa si ambos máximos son iguales?

Los dos máximos de

```python
def maximo(lista):
    max_ = -1
    for i in lista:
        if i > max_:
            max_ = i
    return max_

milista = [20,20,50,70,50,30,70,42,42,50]
# Encontrando el primer mayor
maximo_1 = maximo(milista)
milista.remove(maximo_1)
# Encontrando el segundo
maximo_2 = maximo(milista)
```

Salida

```
70 70
[20,20,50,50,30,70,42,42,50]
```

No es la única forma de encontrar los dos mayores

Existencia e Índice

Muchas veces tenemos listas que tienen tantos elementos, que no sabemos si algún elemento existe en una lista e incluso en qué posición de la lista se encuentra.

¿Cómo podríamos hacerlo?

Existencia

Podríamos revisar la lista completa y compararla con el valor.

¿Cómo se haría?

```
def existe(lista, valor):
    existe = False
    for i in lista:
        if i == valor:
            existe = True
    return existe

milista = [20,20,50,70,50,30,70,42,42,50]
print(existe(milista, 43))
print(existe(milista, 50))
```

Existencia mejorado

```
def existemejor(lista, valor):
    if valor in lista:
        return True
    else:
        return False

milista = [20,20,50,70,50,30,70,42,42,50]
print(existe(milista, 43))
print(existe(milista, 50))
```

Existencia super mejorado

```
milista = [20,20,50,70,50,30,70,42,42,50]
print(43 in milista)
print(50 in milista)
```

Índice

¿Cómo podemos saber en qué posición se encuentra un elemento en la lista?

¿Qué pasa si el elemento está más de una vez?

```
def indice(s, lista):
    for i in range(0, len(lista)):
        if s == lista[i]
            return i
    return -1
```

Index

```
def indice(s, lista):
    for i in range(0, len(lista)):
        if s == lista[i]
            return i
    return -1
```

```
x = lista.index(elemento)
print(x)
```

Ordenamiento

Muchas veces vamos a necesitar ordenar los datos antes de procesarlos y/o mostrarlos

Vamos a empezar por el caso más simple: ordenar una lista de números enteros

Clase 08: Algoritmos con listas

Ordenando una lista

```
lista = [5, 8, 2, 6, 3, 9]
print(lista)

for a in range(len(lista) - 1):
    for b in range(a + 1, len(lista)):
        if lista[a] > lista[b]:
            aux = lista[a]
            lista[a] = lista[b]
            lista[b] = aux

print(lista)
```

```
[5, 8, 2, 6, 3, 9]
[2, 3, 5, 6, 8, 9]
```

Fíjate que lo que estamos haciendo es comparar cada elemento contra los que vienen “después”

Y hay un **if** que permite determinar cuándo es necesario intercambiar los elementos.

En este caso, el orden es de menor a mayor, por lo que el **if** detecta cuando el primer elemento es mayor, y por lo tanto necesita intercambiarse

Algunas consideraciones

El primer elemento se compara contra todos los otros elementos

Pero a medida que avanza el primer índice, la cantidad de comparaciones va disminuyendo

El segundo ciclo comienza en a+1, ya que los elementos que están antes de ese índice ya están ordenados!

Algunas consideraciones

```
lista = [5, 8, 2, 6, 3, 9]
print(lista)

for a in range(len(lista) - 1):
    for b in range(a + 1, len(lista)):
        if lista[a] > lista[b]:
            aux = lista[a]
            lista[a] = lista[b]
            lista[b] = aux

print(lista)
```

Debido a este **if**, ¿cómo se ordena la lista?

Si se quiere ordenar en otro sentido, ¿qué hay que hacer?

Dos listas

Muchas veces nos va a suceder que tenemos dos listas, y ambas están relacionadas.

Por ejemplo:

```
nombres = ['Juan', 'Pedro', 'María']
edades = [15, 11, 19]
```

¿Qué podemos hacer para escribir el nombre de las personas, pero ordenadas por edad de mayor a menor?

Dos listas

Al momento de determinar que es necesario intercambiar los elementos (en el **if**), si movemos una lista, hay que mover la otra también.

De esta forma se mantiene la estructura, y las listas siguen siendo paralelas

```
nombres = [ "Juan", "Pedro", "María" ]
edades = [ 15, 11, 19]

def imprimir(mensaje, lista1, lista2):
    print(mensaje)
    for i in range(len(lista1)):
        print(lista1[i], lista2[i])

imprimir("inicial", nombres, edades)

for a in range(len(edades) - 1):
    for b in range(a + 1, len(edades)):
        if edades[a] < edades[b]:
            aux = edades[a]
            edades[a] = edades[b]
            edades[b] = aux
            aux = nombres[a]
            nombres[a] = nombres[b]
            nombres[b] = aux

imprimir("final", nombres, edades)
```

Tres listas

En este caso, ¿qué criterio estamos usando para ordenar?

¡Podemos mejorar el **if**!

```
nombres = [ "Juan", "Pedro", "María" ]
edades = [ 15, 11, 19]
estatura = [ 170, 150, 165 ]

def imprimir(mensaje, lista1, lista2, lista3):
    print(mensaje)
    for i in range(len(lista1)):
        print(lista1[i], lista2[i], lista3[i])

imprimir("inicial", nombres, edades, estatura)
for a in range(len(estatura) - 1):
    for b in range(a + 1, len(estatura)):
        if estatura[a] < estatura[b]:
            aux = edades[a]
            edades[a] = edades[b]
            edades[b] = aux
            aux = nombres[a]
            nombres[a] = nombres[b]
            nombres[b] = aux
            aux = estatura[a]
            estatura[a] = estatura[b]
            estatura[b] = aux
imprimir("final", nombres, edades, estatura)
```

Tres listas

¡Mucho más fácil de leer!

```
nombres = [ "Juan", "Pedro", "María" ]
edades = [ 15, 11, 19]
estatura = [ 170, 150, 165 ]

def imprimir(mensaje, lista1, lista2, lista3):
    print(mensaje)
    for i in range(len(lista1)):
        print(lista1[i], lista2[i], lista3[i])

def intercambiar(lista, indice1, indice2):
    aux = lista[indice1]
    lista[indice1] = lista[indice2]
    lista[indice2] = aux

imprimir("inicial", nombres, edades, estatura)

for a in range(len(estatura) - 1):
    for b in range(a + 1, len(estatura)):
        if estatura[a] < estatura[b]:
            intercambiar(edades, a, b)
            intercambiar(nombres, a, b)
            intercambiar(estatura, a, b)

imprimir("final", nombres, edades, estatura)
```

Ordenamiento multicriterio

```
nombres = ['Juanita', 'Pedro', 'María', 'José', 'Ximena']
edades  = [   40    ,   25   ,   25   ,   25  ,    30   ]
```

Ahora queremos escribir por pantalla a las personas, ordenadas por su edad

Fíjate que hay personas que tienen la misma edad. En ese caso queremos escribir los nombres ordenados de forma alfabética

¿Cómo hacemos para que esto suceda?

José 25

María 25

Pedro 25

Ximena 30

Juanita 40

Ordenamiento multicriterio

```
for a in range(len(nombres) - 1):
    for b in range(a + 1, len(nombres)):
        if edades[a] > edades[b]:
            intercambiar(edades, a, b)
            intercambiar(nombres, a, b)
        elif edades[a] == edades[b]
            if nombres[a] > nombres[b]:
                intercambiar(edades, a, b)
                intercambiar(nombres, a, b)
```

Fíjate que en este caso, si al comparar dos elementos, éstos tienen la misma edad, pasamos a comparar por el siguiente criterio para ver si requiere que los elementos se intercambien

Trabajo autónomo mínimo

Revisar capítulo 9(1) libro guía.

Resolver ejercicios 1, 2, 3 .