

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN



Diseño y análisis de algoritmos

Tarea 2.- Búsqueda lineal

Profesor: Jesus Hernandez Cabrera

Alumno: Gerson Neftaly Juarez Lopez

Grupo : 1558

Como podemos ver el programa funciona para ellos utilizamos los siguientes casos:

```
casos = [
    {"array": [1,2,8,9,6,19,3,4,300],
     "numero": 19},
    {"array": [1,2,8,9,6,19,3,4,300,600,434,1234,4335,3,0,132321,33322,34444332,22333244],
     "numero": 22333244},
]
```

lo cual nos retorna los siguiente:

```
gerso@neft_neft MINGW64 ~/Downloads/algoritmos
$ C:/Python313/python.exe c:/Users/gerso/Downloads/algoritmos/tarea2.py
Se busca el numero 19 en el array [1, 2, 8, 9, 6, 19, 3, 4, 300]
el numero ha sido hallado en la posicion 5 y se han realizado 6 comparaciones
Se busca el numero 22333244 en el array [1, 2, 8, 9, 6, 19, 3, 4, 300, 600, 434, 1234, 4335, 3, 0, 132321, 33322, 3
4444332, 22333244]
el numero ha sido hallado en la posicion 18 y se han realizado 19 comparaciones
```

por lo cual nuestro programa funciona.

Ahora calcularemos $T(n)$ y $S(n)$:

Primero calculemos $T(n)$:

```
2 def busquedaLineal(arreglo, numeroABuscar):
3     comparaciones = 0
4     for i in range(len(arreglo)):
5         comparaciones += 1
6         if arreglo[i] == numeroABuscar:
7             return comparaciones, i
8     return comparaciones, -1
```

comparaciones -> 1

Luego si miramos dentro del for notaremos que se hacen dos operaciones por iteracion al hacer el if y al sumar uno a comparaciones.

El return se ejecuta una sola vez, por lo que es solo una operación si es que se ejecuta, lo mismo sucede con el ultimo return, es solo una operación si es que se ejecuta.

Por lo que:

$$2n + 1 + 1 + 1 = 2n + 3$$

```
1. comparaciones = 0          → 1 operación
2. for i in range(n)          → 1 operación (crear range)
3. n veces:
    - comparaciones += 1      → n operaciones
    - if arreglo[i] == valor  → n operaciones
4. return comparaciones, -1   → 1 operación
```

TOTAL: $1 + 1 + n + n + 1 = 2n + 3$ operaciones

Calculamos $S(n)$:

```
def busquedaLineal(arreglo, numeroABuscar):
    comparaciones = 0          # 1 variable (entero)
    for i in range(len(arreglo)): # 1 variable (iterador)
        comparaciones += 1      #
        if arreglo[i] == numeroABuscar: #
            return comparaciones, i #
    return comparaciones, -1    #
```

por lo que $S(n) = 2$