

Análisis de métodos de buscar

Diego Quirós Artiñano

Universidad Nacional de Costa Rica

EIF-203: Estructuras Discretas (10 A.M.)

Carlos Loria-Saenz

24 de marzo, 2022



Índice

Índice de figuras

Índice de tablas

Análisis de es_simetrica()

0. El algoritmo original.

```
def es_simetrica(a: list[list[int]]) -> bool:
    for i in range(len(a)):
        for j in range(i+1, len(a[i])):
            if i != j and a[i][j] != a[j][i]:
                return False
    return True
```

1. Tamaño de los datos

El método se construyó de manera en la que no busque la diagonal de la matriz y solo se tiene que buscar la parte de arriba, entonces el tamaño de datos es

$$\frac{n(n-1)}{2}$$

2. Operaciones de Interés

| Notación [en función de tiempo (T)] | Operación | Tipo de Operación |
|-------------------------------------|--|------------------------|
| $T_{range()}$ | range(len(a)) or range(i+1, len(a[i])) | constante |
| $T_{!=}$ | i != j or a[i][j] != a[j][i] | constante |
| $T_{[]}$ | a[i][j] or a[j][i] | constante [1ex] height |

Cuadro 1

Operaciones es_simetrica()

Peor caso: toas son iguales a la mas grande de todas, como supuesto

Asumimos que vale 1 (unidad de tiempo), según el más alto

3. Ecuación

```

# Separando por partes
# Parte 1
p:int = 0
# Parte 2
while p < len(a):
    if x == a[p]:
        return p
    else:
        p += 1
# Parte 3
return -1

```

$$T_{\text{buscar_while}}(n) = T_{\text{parte1}} + T_{\text{parte2}} + T_{\text{parte3}}$$

$$T_{\text{buscar_while}}(n) = 1 + T_{\text{while}}(n) + 0$$

$$T_{\text{buscar_while}}(n) = T_{\text{while}}(n) + 1$$

Evaluando $T_{\text{while}}(n)$:

2 operaciones: <y len(a)

2 operaciones: == y []

1 operación: +=

Entonces:

$$T_{\text{while}}(n) = 2 + 2 + 1 + T_{\text{while}}(n - 1)$$

$$T_{\text{while}}(n) = 5 + 5 + 5 + \dots T_{\text{while}}(0)$$

Como el while va a verificar si sigue siendo parte del while entonces se cuentan las primeras 2 operaciones otra vez:

$$T_{\text{while}}(N) = 5n + 2$$

4. Volviendo a meter en la ecuación original:

$$T_{\text{buscar_while}}(n) = 5n + 2 + 1$$

$$T_{\text{buscar_while}}(n) = 5n + 3$$

5. Orden de crecimiento

Como la ecuación es una función lineal entonces el orden de crecimiento también

lo cual significa que: $O(n) \sim T_{\text{buscar_while}}(n)$

6. Código

```
def buscar_while_instrumentado(x:int , a:list[int]) -> int:
    contador_operaciones = 0
    p:int = 0
    contador_operaciones += 1
    no_salir = p < len(a)
    contador_operaciones += 2
    while no_salir:
        if x == a[p]:
            contador_operaciones += 2
            return contador_operaciones
        else:
            contador_operaciones += 2
            p += 1
            contador_operaciones += 1
        no_salir = p < len(a)
        contador_operaciones += 2
    return contador_operaciones
```

Excel

Para importar a Excel para hacer la gráfica, se hizo este código:

```
def test_buscar_instrumentado(filename, init, maxi, inc):  
    file = open(filename, 'w')  
    file.write('n;time_while;time_for\n')  
    for n in range(init, maxi, inc):  
        a = list(range(n))  
        x = n  
        file.write(f'{n};{buscar_while_instrumentado(x,a)};{buscar_for_instrumentado(x,a)}\n')  
    file.close()  
test_buscar_instrumentado('buscar_instrumentado.csv', 10, 200, 10)
```

Figura 1

Gráfica de los métodos en Excel