

¿Qué es el análisis de algoritmos?

Es el proceso de analizar el rendimiento de los algoritmos en función del tiempo de ejecución y el uso de espacio.

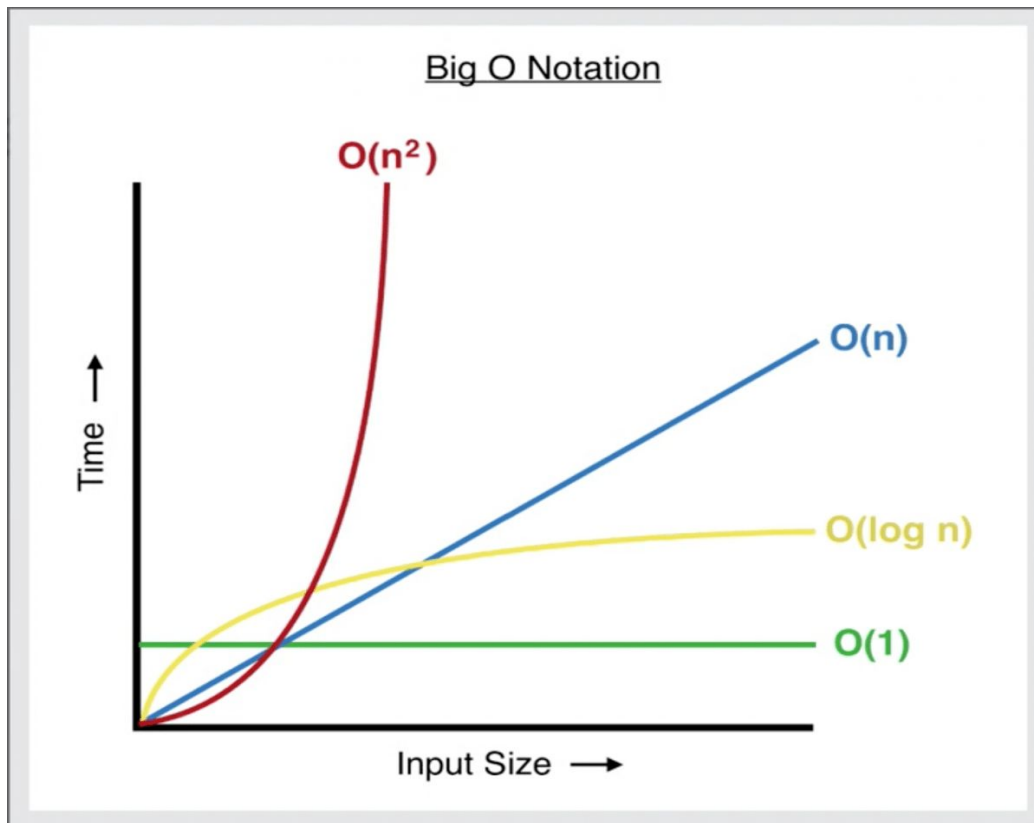
Conceptos importantes

Notación Big o: representa el tiempo de ejecución en el peor de los casos.

Eficiencia temporal: tiempo de ejecución de un algoritmo en función del tamaño de entrada.

Eficiencia espacial: espacio que ocupa un algoritmo en función del tamaño de entrada

Notación Big O



Caso Práctico

```
import time
```

```
def fibonacci_recursivo(n):
```

```
    if n <= 1:
```

```
        return n
```

```
    return fibonacci_recursivo(n-1) +  
    fibonacci_recursivo(n-2)
```

```
def fibonacci_iterativo(n):
```

```
    a, b = 0, 1
```

```
    for _ in range(n):
```

```
        a, b = b, a + b
```

```
    return a
```

```
def medir_tiempo(funcion, n):
```

```
    inicio = time.time()
```

```
    resultado = funcion(n)
```

```
    fin = time.time()
```

```
    return resultado, fin - inicio
```

```
if __name__ == "__main__":
```

```
    n = 42
```

```
    resultado1, tiempo1 = medir_tiempo(fibonacci_iterativo, n)
```

```
    resultado2, tiempo2 = medir_tiempo(fibonacci_recursivo, n)
```

```
    print(f"Fibonacci Iterativo: Resultado={resultado1},  
    Tiempo={tiempo1:.8f} segundos")
```

```
    print(f"Fibonacci Recursivo: Resultado={resultado2},  
    Tiempo={tiempo2:.8f} segundos")
```

Resultados Obtenidos

- Ambos algoritmos devuelven el mismo resultado correcto.
- El algoritmo iterativo de fibonacci es notablemente más rápido que el recursivo para valores grandes de n . La fórmula recursiva tarda cientos o miles de veces más que la iterativa debido a su complejidad exponencial ($O(2^n)$) frente a la lineal ($O(n)$) del método iterativo.
- El tiempo de ejecución para $n = 42$ en fibonacci recursivo fue cientos o incluso miles de veces mayor que con el método iterativo.

Conclusión

El análisis de algoritmos nos permite evaluar el rendimiento de los algoritmos considerando dos factores importantes: el tiempo de ejecución y el consumo de memoria. Esta evaluación resulta sustancial cuando existen múltiples algoritmos capaces de resolver un mismo problema, ya que nos brinda criterios para seleccionar la opción más eficiente y adecuada a nuestros requerimientos.