UNIVERSIDAD NACIONAL DE INGENIERÍA

Facultad de Ciencias

Ciencia de la Computación

PROGRAMACION PARALELA - CC332B

Tarea 1 - Complejidad de Problemas Computacionales

Elaborado por:

César Omar López Arteaga

Profesor:

Raul Eduardo Huarote Zegarra

Septiembre del 2025

Contenido

Pregunta 1 **2**

Pregunta 2 **4**

Pregunta 1

Identificar y calcular la complejidad computacional O(n), de método de Newton-Raphson de alta dimensión al nivel matemático.

Solucion:

Se analiza la complejidad computacional del método de Newton-Raphson para resolver el sistema

$$F(x) = 0, \qquad F : \mathbb{R}^n \to \mathbb{R}^n.$$

El coste por iteración se descompone en tres partes principales:

1. Evaluación de $F(x)$,
2. Montaje (o aproximación) de la Jacobiana $J(x)$,
3. Resolución del sistema lineal $J(x)\,\Delta x = -F(x)$.

Asi,sea n la dimensión del problema y k el número de iteraciones de Newton necesarias para alcanzar una precisión ε. Nuestro objetivo es estimar la complejidad total en función de n (y, cuando sea relevante, de la estructura de J).

Comenzaremos a calcular los costes por iteración (caso denso o mas critico):

- Evaluación de $F(x)$: Cada componente $f_i(x)$ se evalúa en tiempo $O(1)$, y como hay n componentes, el coste total es

$$O(n).$$

- Montaje de la Jacobiana $J(x)$: La Jacobiana es una matriz $n \times n$. Construirla explícitamente requiere en general

$$O(n^2)$$

operaciones (cálculo de n^2 entradas). Con diferencias finitas, el coste puede estimarse como $O(n \cdot C_f)$, donde C_f es el coste de evaluar F. **- Resolución del sistema lineal:**

El sistema:

$$J(x)\,\Delta x = -F(x)$$

se resuelve típicamente mediante factorización LU. Si J es densa, el coste es

$$O(n^3).$$

- Conclusión por iteración: El coste total de una iteración es

$$O(n) + O(n^2) + O(n^3) = O(n^3).$$

Asi la complejidad total considerando k iteraciones requeridas es:

$$O(k\,n^3).$$

Dado que k suele ser pequeño (por convergencia cuadrática de Newton), en la práctica la complejidad dominante es $O(n^3)$.

Por lo tanto para el problema mas desfavorable (denso), el coste por iteración $O(n^3)$, coste total $O(kn^3) \approx O(n^3)$.

Preguta 2

¿Cuánto tiempo se requiere para obtener el 90 valores de la secuencia de Fibonacci para ambos programas?

Solucion:

Se observa que el método Iterativo realiza el procesamiento computacional de manera óptima con tiempo de procesamiento de complejidad lineal, lo cual es mas rapido el procesamiento. Por otro lado, el método recursivo realiza el procesamiento computacional de la secuencia de Fibonacci con tiempo de procesamiento que se incrementa de manera exponencial a medida que aumenta el valor de n, esto hace el procesamiento mas lento. Asi tenemos, los resultados siguientes:

PARA PROGRAMACION ITERATIVA:

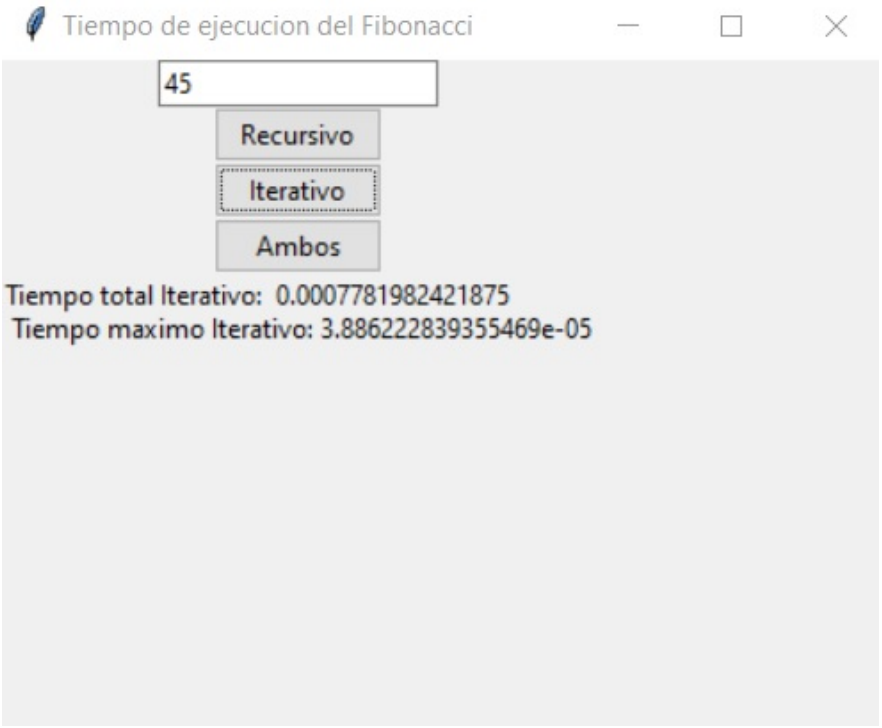

Figure 1: Resultado para el calculo de los 45 terminos, usando el metodo iterativo

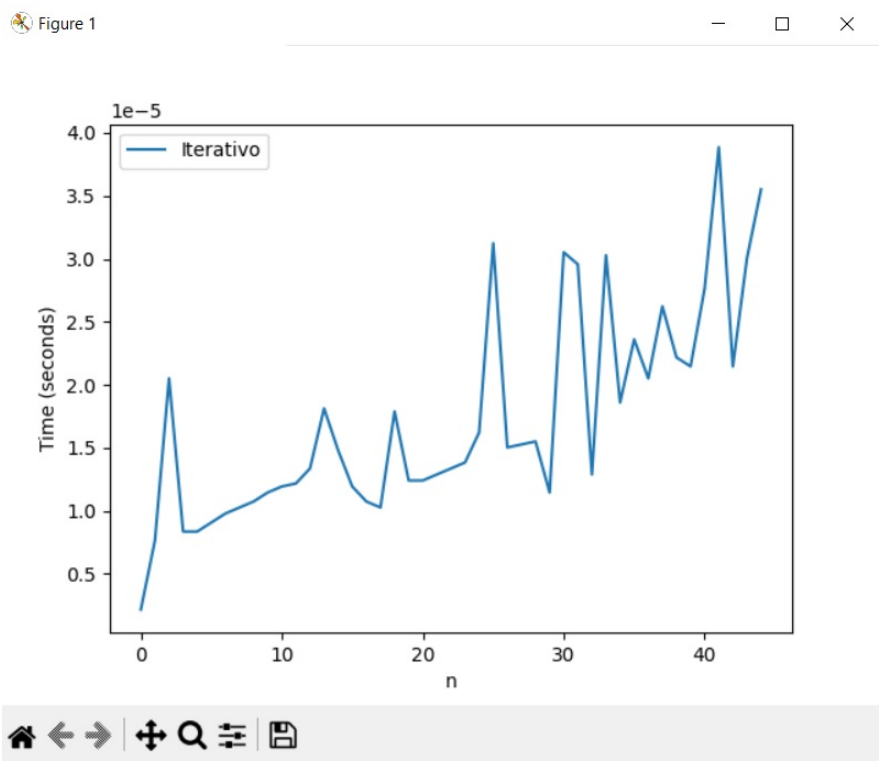

Figure 2: Grafico del tiempo requerido para el metodo iterativo.

PARA PROGRAMACION RECURSIVA:

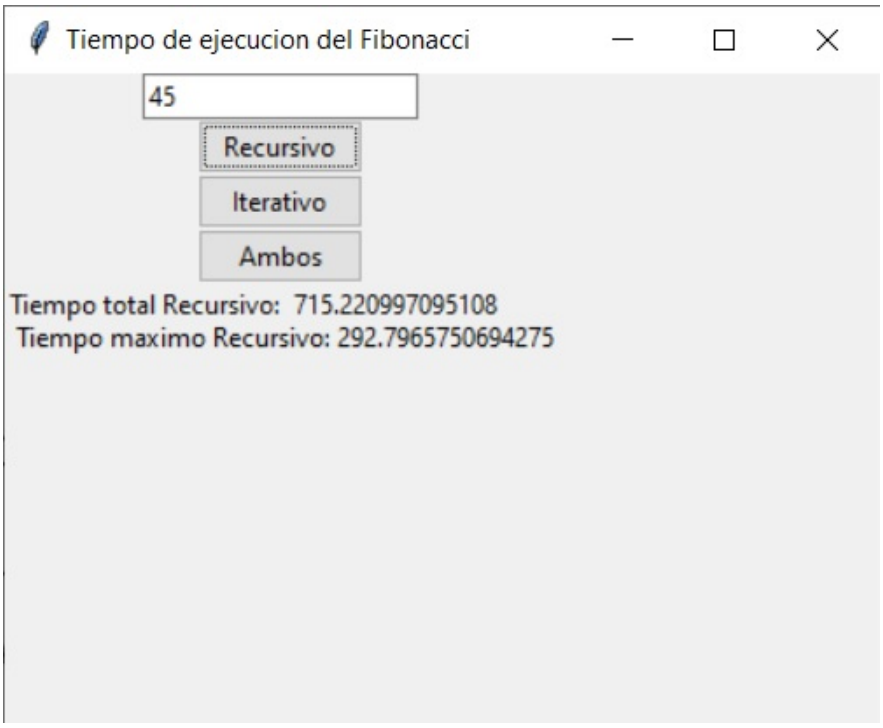

Figure 3: Resultado para el calculo de los 45 terminos, usando el metodo recursivo

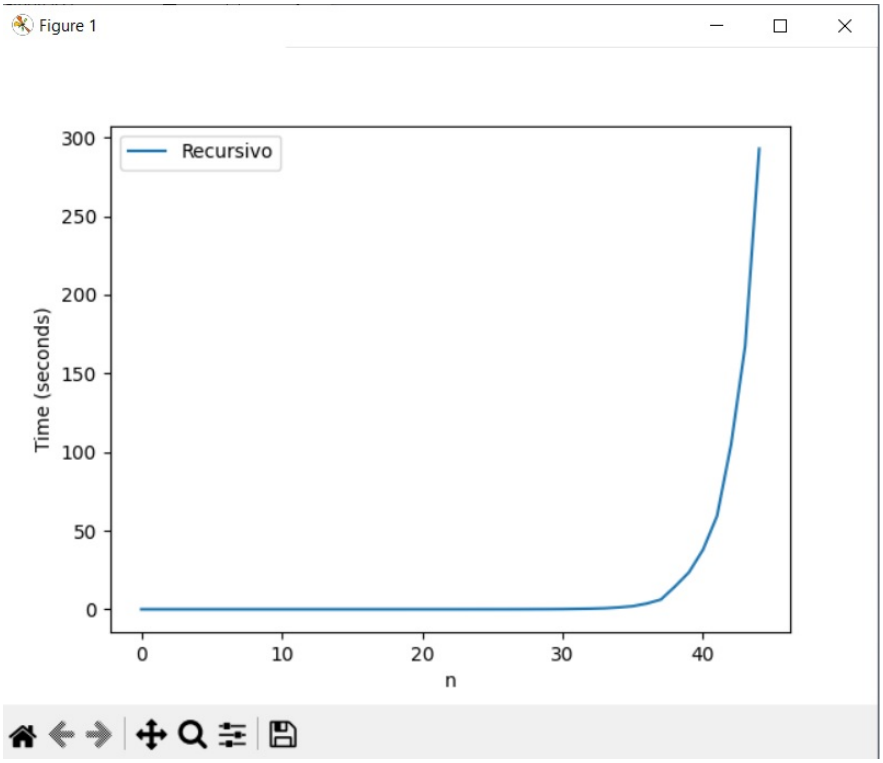

Figure 4: Grafico de tiempos requeridos para el metodo recursivo.

PARA AMBOS METODOS DE PROGRAMACION ITERATIVO-RECURSIVO:

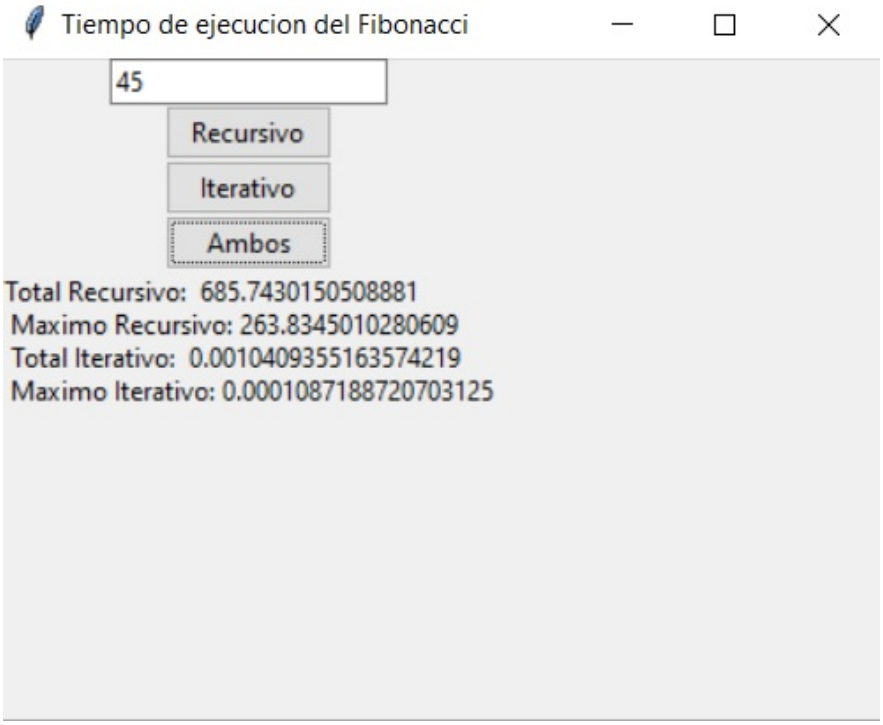

Figure 5: Resultado para el calculo de los 45 terminos, usando ambos metodos

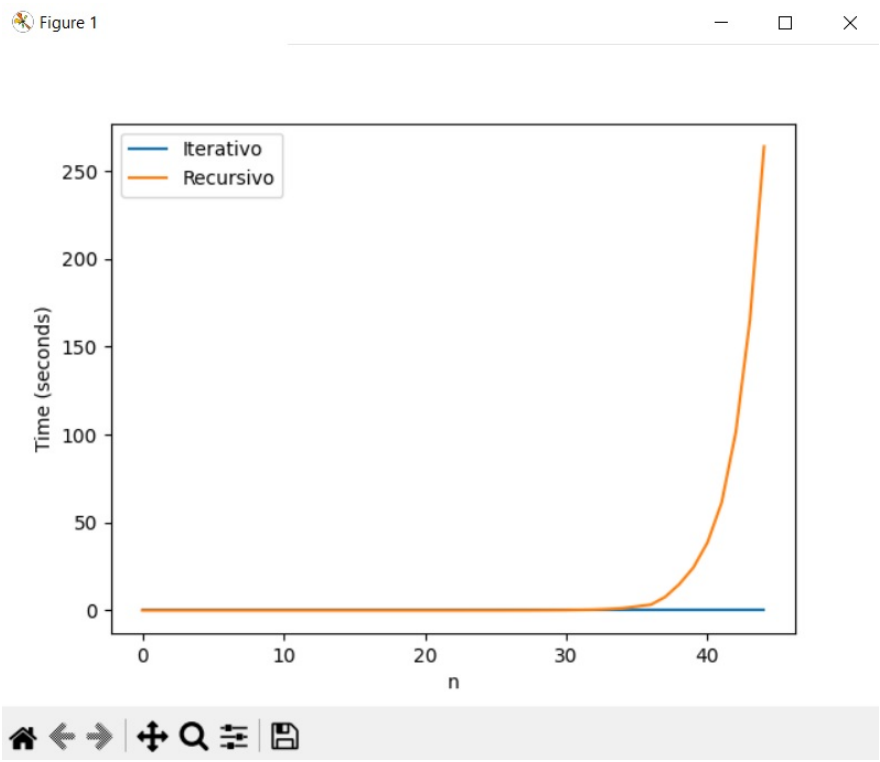

Figure 6: Grafico de tiempos requeridos para ambos metodos.

RESUMEN DE RESULTADOS

- Tiempo para 45 valores de manera recursiva (milisegundos) =685.743

- Tiempo para 45 valores de manera iterativa (milisegundos) = 0.0010409