

COMPLEJIDAD DE ALGORITMOS (CC76)

**Ciclo 2020-2**

**DD1**

**PREGUNTA 1 (5 puntos) – Comparación de tiempos de ejecución**

Para cada función f(n) y tiempo t en la siguiente tabla, determine el mayor tamaño de n de un problema que puede ser resuelto en tiempo t, asumiendo que el algoritmo para resolver el problema toma f(n) microsegundos.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **1 segundo** | **1 minuto** | **1 hora** | **1 día** | **1 mes** | **1 año** | **1 siglo** |
| **lg(n)** |  |  |  |  |  |  |  |
| **√n** |  |  |  |  |  |  |  |
| **n** | **10^6** | **6\*10^7** | **36\*10^8** | **8.24\*10^10** | **2.592\*10^12** | **3.1104\*10^13** | **1.1104\*10^15** |
| **n lg(n)** |  |  |  |  |  |  |  |
| **n2** |  |  |  |  |  |  |  |
| **n3** |  |  |  |  |  |  |  |
| **2n** |  |  |  |  |  |  |  |
| **n!** |  |  |  |  |  |  |  |

**PREGUNTA 2 (10 puntos)**

**– Tiempos de Ejecución Asintóticos**

Indique el tiempo de ejecución asintótico utilizando la notación Big O

1. Algoritmo para hallar el par de puntos más próximo (1 punto)

|  |  |  |
| --- | --- | --- |
| **Línea** | **Pseudocódigo** | **Complejidad** |
| 01 | ALGORITMO ParMasProximo | ----------- |
| 02 | ENTRADA: P (conjunto de puntos), n (tamaño de P) | ----------- |
| 03 | SALIDA : El par de puntos más próximo | ----------- |
| 04 | minDist = ∞ | O(1) |
| 05 | PARA i DESDE 1 HASTA n - 1 HAGA | O(n) |
| 06 | PARA j DESDE i + 1 HASTA n HAGA | O(n/2) |
| 07 | p = P[i], q = P[j] | O(2) |
| 08 | SI dist(p, q) < minDist ENTONCES | O(2) |
| 09 | minDist = dist(p, q) | O(1) |
| 10 | parMasProximo = (p, q) | O(1) |
| 11 | RETORNA parMasProximo | O(1) |
| 12 | FINALGORITMO | ----------- |
| **Análisis Asintótico - Peor Caso (Big O)** | | O(n^2) |

1. Algoritmo para hallar un valor en un arreglo. Justifique su solución. (2 puntos)

|  |  |  |
| --- | --- | --- |
| **Línea** | **Pseudocódigo** | **Complejidad** |
| 01 | ALGORITMO Busqueda | ----------- |
| 02 | ENTRADA: A (arreglo de valores), p (inicio),  r (fin), V (valor deseado) | ----------- |
| 03 | SALIDA : Índice de V o -1 si no se encuentra V | ----------- |
| 04 | SI r >= p ENTONCES | O(1) |
| 05 | q1 = p + (r - p) / 4 | O(4) |
| 06 | q2 = 2 \* q1 | O(2) |
| 07 | q3 = 3 \* q1 | O(2) |
| 08 | SI A[q1] == x ENTONCES RETORNA q1 | O(3) |
| 09 | SINO SI A[q2] == x ENTONCES RETORNA q2 | O(3) |
| 10 | SINO SI A[q3] == x ENTONCES RETORNA q3 | O(3) |
| 11 | SI A[q1] > x ENTONCES  RETORNA Busqueda(A, p , q1 - 1, x) | O(logn) |
| 12 | SINO SI A[q2] > x ENTONCES  RETORNA Busqueda(A, q1, q2 - 1, x) | O(logn) |
| 13 | SINO SI A[q3] > x ENTONCES  RETORNA Busqueda(A, q2, q3 - 1, x) | O(logn) |
| 14 | SINO ENTONCES  RETORNA Busqueda(A, q3, r , x) | O(logn) |
| 15 | SINO ENTONCES RETORNA -1 | O(1) |
| 16 | FINALGORITMO | ----------- |
| **Análisis Asintótico - Peor Caso (Big O)** | | O(logn) |

1. Si modificara el código anterior para aceptar n particiones del arreglo (2 ≤ n ≤ Tamaño del arreglo). (2 puntos)
2. ¿Cuál sería el límite asintótico para cualquier n? (justifique su respuesta)
3. ¿Qué sucede cuando n es igual al tamaño del arreglo?

**PREGUNTA 3 (5 puntos)**

Usando el teorema maestro indique Big O para las siguientes recurrencias:

1. **T(n) = 9T(n/3) + n : O(nlog3(1))**
2. **T(n) = T(2n/3) + 1 : O(n0log(n))**
3. **T(n) = 3T(n/4) + n lg n :** **O(n1)**
4. **T(n) = 2T(n/2) + n lg n : O(n1log(n))**