Primer examen opcional

Fundamentos de análisis y diseño de algoritmos

Carlos Andres Delgado S, Ing *

10 de Mayo de 2018

Nombre:______________________________
Código:_______________

1. Complejidad computacional e iterativa [50 puntos]

Para el siguiente algoritmo:

```
1  algoritmo(n)
2
3      for(i=0; i<2n; i++)
4
5          j = i
6          s = n
7
8          while(j<n)
9              s+=4
10             j++
11         end
12
13         k = 0
14         t = 2
15         while(k<=i)
16             t=4*t
17             k++
18         end
19     end
```

1. **(25 puntos)** Calcule la complejidad total del algoritmo. Muestre el procedimiento línea por línea. Finalmente, indique la complejidad total en términos de $O(f(n))$ siendo $f(n)$ la cota más pequeña posible.

2. Para el ciclo interno de las lineas (8 y 11) y el ciclo interno de las lineas (15-18)

   - **(5 puntos)** Forma de estado, estado inicial, transformación de estados y estado final, para cada uno de los ciclos internos.
   - **(20 puntos)** Invariante de ciclo y su demostración. Para cada uno de los ciclos internos.

   Aunque al ciclo externo no se le va calcular la invariante de ciclo, tome en cuenta cómo los valores que toma i afectan a los ciclos internos.

2. Diseño de soluciones [50 puntos]

2.1. Diseño iterativo [25 puntos]

Problema Inversiones de un arreglo: Sea $A[1...n]$ un arreglo n números distintos. Si i ¡j y $A[i] < A[j]$, entonces el par (i, j) es llamado una inversión de A

1. **(2 puntos)** Dado el arreglo $[5, 9, 12, 10, 3]$ indique todas sus inversiones

2. **(3 puntos)** ¿Cómo debe ser el arreglo de tal forma de obtengan el mayor número de inversiones?

3. **(5 puntos)**Explique como es la solución ingenua de este algoritmo. Indique su complejidad.

4. **(15 puntos)** Escriba un pseudocódigo de un algoritmo iterativo para **contar el número** de inversiones de un arreglo A cualquiera. Indique la **invariante de ciclo** y calcule la complejidad de este algoritmo.

1

2.2. Diseño divide y vencerás [25 puntos]

Problema de dominancia: Dado un par de puntos (x, y), (x', y') en el plano, se dice que (x, y) domina a (x', y') si $x > x'$ y $y > y'$. También se dice que (x', y') es dominado por (x, y).

Entrada: Un conjunto de puntos en el plano $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n\}$ tales que x_i ¡x_{i+1} o $(x_i = x_{i+1} \wedge y_i < y_{i+1})$. Es decir, los n puntos vienen ordenados de izquierda a derecha y de abajo hacia arriba.

Salida: Un conjunto de puntos que no están dominados

1. **(5 puntos)** Explique como es la solución ingenua de este algoritmo. Indique su complejidad.

2. **(7 puntos)** Explique como un dibujo como es la estrategia de solución de este problema con divide y vencerás. Utilice un ejemplo con 5 puntos

3. **(13 puntos)** Escriba un pseudocódigo que utilice la estrategia divide y vencerás para solucionar este problema. Calcule la complejidad de esta solución. Resuelva la ecuación de recurrencia asociada con el método del maestro.

*carlos.andres.delgado@correounivalle.edu.co

Ayudas

Sumatorias

$\sum_{k=1}^{n} c = cn \quad \sum_{k=1}^{n} k = \frac{n(n+1)}{2} \quad \sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$

$\sum_{k=0}^{n} ar^k = \frac{ar^{(n+1)}-a}{r-1}$ Si $r \neq 1$

$\sum_{k=0}^{n} ar^k = (n+1)a$ Si $r = 1$

Formulas solución método del maestro

Recuerde la forma $T(n) = aT(\frac{n}{b}) + f(n)$

- Si $f(n) = O(n^{log_b a - \epsilon})$ para algún $\epsilon > 0$ entonces $T(n) = \Theta(n^{log_b a})$
- Si $f(n) = \Theta(n^{log_b a})$ entonces $T(n) = \Theta(log(n) * n^{log_b a})$
- Si $f(n) = \Omega(n^{log_b a + \epsilon})$ para algún $\epsilon > 0$ y existe un $c < 1$ tal que $af(\frac{n}{b}) <= cf(n)$ entonces $T(n) = \Theta(f(n))$.