Universidad del Valle

Primer examen opcional
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS
Grupo 02
Duración: 2 horas

Carlos Andres Delgado S, Ing *

15 de Abril de 2015

1. Computación iterativa y complejidad algoritmos [45 puntos]

1.1. Complejidad algoritmos [20 puntos]

Para el siguiente algoritmo:

```
Algoritmo(int N)
{
        int i,res;
        i = 2;
        res = 10;

        while(i<N){

                res = res + 2*i;

                for(int j=0; j<=(i+1); j++)
                {
                        res = res + 1;
                }
                i++;
        }
        System.out.println“(Resultado“= + Res);
}
```

(20 puntos) Muestre cuantas veces se ejecuta cada línea del código en términos de n y dé el total de ejecuciones del algoritmo en términos de n. Indique la complejidad del algoritmo en términos de $O(f(n))$.

1.2. Computación interativa [25 puntos]

Diseñe un algoritmo utilizando ciclos que calcule para entrada N una salida igual a $\sum_{i=0}^{n} 3i$

1. (5 puntos) Escriba el algoritmo que soluciona este problema.
2. (5 puntos) ¿Cómo puede representar los estados del algoritmo?. ¿Cual es el estado inicial?.
3. (7 puntos) ¿Cómo es la transición de estados del algoritmo?.
4. (8 puntos) ¿Cual es la invariante de ciclo del algoritmo?.

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

2. Crecimiento de funciones [20 puntos]

1. (8 puntos) Demuestre que $n + 2$ es $\Theta(n)$.
2. (12 puntos) Indique si existen funciones $f(n)$ y $g(n)$ tales que $f(n)$ es $o(g(n)$ y $g(n)$ es $\Theta(f(n))$. En caso de existir dé un ejemplo de funciones $f(n)$ y $g(n)$

3. Ecuaciones de recurrencia [15 puntos]

Para las siguientes preguntas asuma que $T(1) = 4$.

1. (10 puntos) Con el método de iteración solucione $T(n) = 4T(\frac{n}{2}) + \frac{n}{4} + 2$. Exprese en forma de sumatorias.
2. (5 puntos) Con el método del maestro determine la solución a la siguiente ecuación de recurrencia $T(n) = 8T(\frac{n}{16}) + n$.

4. Estructuras de datos [20 puntos]

Indicar cuales de las siguientes expresiones son verdaderas y falsas, en caso de ser falsas justifique su respuesta.

1. Estructura **Pilas (Stack)** y **Colas (Queue)**
   a) [**4 puntos.**] Una cola se puede considerar una estructura $FIFO$ es decir, que el primer elemento encolado es el primero en ser desencolado.
   b) [**4 puntos.**] La complejidad de las operaciones **Push** y **Pop** en una pila es $\Theta(nlog(n))$
2. Estructura Listas
   a) [**4 puntos.**] En la lista **simplemente enlazada** la operación $LIST-INSERT(L, x)$ inserta x al inicio de la lista
   b) [**4 puntos.**] En una lista **doblemente enlazada** la complejidad de $LIST - DELETE(L, x)$ será $\mathcal{O}(1)$
3. Estructura Tablas Hash
   a) [**4 puntos.**] Si se utiliza una buena función hash en una tabla hash con un numero de **slots m** y un universo de llaves de tamaño k la complejidad de una **búsqueda**(*exitosa o no*) es $\Theta(n)$

Ayudas

Formulas de sumatorias

- $\sum_{k=1}^{n} c = cn$
- $\sum_{k=1}^{n} k = \frac{n(n+1)}{2}$
- $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$
- $\sum_{k=0}^{n} ar^k = \frac{ar^{(n+1)}-a}{r-1}$ 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 $\epsilon > 0$, $c < 1$ y $af(\frac{n}{b}) <= cf(n)$ entonces $T(n) = \Theta(f(n))$.

Recuerde colocar los procedimientos realizados, ya que estos tienen un gran valor en la calificación de cada punto.

Inserción árboles rojinegros

- Caso 1: x(rojo) es un hijo de un padre rojo y el tío de x es rojo, se pintan de negro padre y tío de x, el abuelo de x queda entonces de rojo. x es ahora el abuelo de x
- Caso 2: x(rojo) es un hijo derecho de un padre rojo y el tío de x, y, es ahora negro. Se rota a la izquierda p[x]. x ahora es el padre de x
- Caso 3: x(rojo) es el hijo izquierdo de un padre rojo y el tío es negro. Se cambian los colores de p[x] y p[p[x]]. Se rota a la derecha p[x].

Figura 1: Rotaciones rojinegros