Primer examen parcial
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS
Grupo 51
Duración: 2 horas

Carlos Andres Delgado S, Ing *

25 de Marzo de 2015

1. Computación iterativa y complejidad algoritmos [44 puntos]

Para el siguiente algoritmo:

```
Algoritmo(int N)
{
        int i,res;
        i = -3;
        res = 0;

        while(i<N){

                res = res + 2*i;

                for(int j=0; j<=(N+1); j++)
                {
                        res = res + 2;
                }
                i++;
        }
        System.out.println“(Resultado“= + Res);
}
```

1.1. Entendimos el problema [8 puntos]

- (3 puntos) Determine las salidas para las siguientes entradas $\{1, 2, 4, 5\}$
- (5 puntos) Para un número $n >= -3, n \in \mathbb{N}$ escriba una expresión que permita determinar la salida este algoritmo.

1.2. Analicemos el algoritmo [36 puntos]

- (18 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 $O(f(n))$.
- (4 puntos) ¿Cómo puede representar los estados del algoritmo?. ¿Cual es el estado inicial?.
- (6 puntos) ¿Cómo es la transición de estados del algoritmo?.
- (8 puntos) ¿Cual es la invariante de ciclo del algoritmo?.

2. Crecimiento de funciones [14 puntos]

- (6 puntos) Demuestre que $n^2 + 4n$ es $\Omega(n^2)$.
- (8 puntos) Indique si existen funciones $f(n)$ y $g(n)$ tales que $f(n)$ es $\Theta(g(n)$ y $g(n)$ es $O(f(n))$. En caso de existir dé un ejemplo de funciones $f(n)$ y $g(n)$

3. Ecuaciones de recurrencia [20 puntos]

Para las siguientes preguntas asuma que $T(4) = 2$.

- (10 puntos) Con el método de iteración solucione $T(n) = 4T(\frac{n}{2}) + n^3 + \frac{1}{4}$. Exprese en forma de sumatorias.
- (10 puntos) Con el método del maestro determine la solución a la siguiente ecuación de recurrencia $T(n) = 16T(\frac{n}{4}) + n^2$.

4. Estructuras de datos [22 puntos]

- (6 puntos) Se tiene un array denominado como S inicialmente vacío de tamaño 7 para construir una cola, muestre como se encuentra el array y el valor de la variable $S.head$ y $S.tail$ después de ejecutar las siguientes instrucciones
  1. ENQUEUE(S,4)
  2. ENQUEUE(S,3)
  3. ENQUEUE(S,7)
  4. DEQUEUE(S)
  5. ENQUEUE(S,8)
  6. DEQUEUE(S)
  7. ENQUEUE(S,-3)
- (16 puntos) Utilizando método de la multiplicación con $A = 0{,}265$ para un tabla hash tamaño de arreglo 15 y función para generación de llaves de tipo String con caracteres de 0 a 127:

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

Carácter	Valor	Carácter	Valor
a	97	b	98
c	99	d	100
e	101	f	102
g	103	h	104
i	105	j	106
k	107	l	108
m	109	n	110
o	111	p	112
q	113	r	114
s	115	t	116
u	117	v	118
w	119	x	120
y	121	z	122

Inserte las siguientes palabras y muestre como se calcula la llave y posición en la tabla de cada elemento, dibuje como queda la tabla hash al final de las operaciones:

Palabra	Palabra
ama	ma
papa	pa
car	casa
red	poke
fada	ds
ia	flp
one	piece

5. Punto extra [+0.5 parcial]

Construya un árbol rojinegro insertando los siguientes elementos $\{4, 10, 12, 2, 1, 3, 15, 19\}$ muestre el procedimiento al insertar cada elemento.

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