



Primer examen parcial
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS
Grupo 80
Duración: 2 horas 30 minutos

Carlos Andres Delgado S, Ing *

07 de Mayo de 2014

1. Computación iterativa y complejidad algoritmos [25 puntos]

Para el siguiente algoritmo:

```
Computa3 (int N)
{
    int A, B, i, j;
    A=0;
    i=1;
    while (i<=N){
        B=1;
        j=1;

        while (j<=3)
        {
            B=B*i;
            j++;
        }
        A=A+B;
        i++;
    }
    System.out.println("Resultado" + A);
}
```

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

1.1. Entendimos el problema [5 puntos]

- (2 puntos) Determine las salidas para las siguientes entradas $\{1, 2, 3, 7, 9\}$
- (3 puntos) Para un número $n \geq 0, n \in \mathbb{N}$ escriba una expresión que permita determinar la salida este algoritmo.

1.2. Analicemos el algoritmo [20 puntos]

- (3 puntos) ¿Cómo puede representar los estados del algoritmo?.
- (3 puntos) ¿Cual es el estado inicial?.
- (4 puntos) ¿Cómo es la transición de estados del algoritmo?.
- (6 puntos) ¿Cual es la invariante de ciclo del algoritmo?.
- (4 puntos) Determine el costo en **espacio en memoria** del algoritmo en términos de $O(f(n))$.

2. Crecimiento de funciones [9 puntos]

- (3 puntos) Que significa que $f(n)$ es $\omega(g(n))$.
- (6 puntos) Demuestre que $2n^2 + 6n + 6$ es $\Theta(n^2)$.

3. Ecuaciones de recurrencia [25 puntos]

Para las siguientes preguntas asuma que $T(2) = 1$.

- (8 puntos) Demuestre con el método de iteración que $T(n) = 2T(\frac{n}{2}) + n^2 + 2$ es $O(n^2)$.
- (8 puntos) Utilizando el método del maestro determine cual es la cota superior e inferior más cercana es decir $\Theta(f(n))$ a $T(n) = 6T(\frac{n}{36}) + n^2 + 3$.
- (9 puntos) Utilice el método de sustitución para determinar que $T(n) = 2T(\frac{n}{2}) + n$ es $O(n \log(n))$.

4. Estructuras de datos [55 puntos]

4.1. Conceptos teóricos [15 puntos]

- (2 puntos) ¿Cual es la complejidad de las operaciones PUSH y POP en una pila?
- (5 puntos) ¿Cual es la complejidad de las operaciones en un árbol binario de búsqueda de altura h en el peor caso?
- (3 puntos) Describa la situación donde las operaciones en un árbol binario de búsqueda corresponden al peor caso.
- (5 puntos) ¿Que característica debe tener una buena función hash?

4.2. Ejercicios sobre estructuras [10 puntos]

- (4 puntos) Se tiene un array denominado como S inicialmente vacío de tamaño 6 para construir una pila, muestre como se encuentra el array y el valor de la variable $S.top$ después de ejecutar las siguientes instrucciones

- PUSH(S,4)
- PUSH(S,3)
- PUSH(S,7)
- POP(S)
- PUSH(S,8)
- POP(S)

- (6 puntos) ¿Como implementaría una pila utilizando una lista enlazada? Las operaciones POP y PUSH deberían tomar $O(1)$.

4.3. Aplicación [30 puntos]

4.3.1. Problema

Se requiere crear una aplicación para almacenar la información acerca de la estructura de una organización, se debe almacenar los siguientes datos:

- Los cargos dentro de la organización de menor a mayor rango, estos están codificados con un número que indica su importancia, inicialmente están codificados de la siguiente forma:
 - Presidente se codifica como 100000
 - Vicepresidente se codifica como 95000
 - Jefe de área se codifica como 80000
 - Jefe de grupo se codifica como 70000
 - Empleado experimentado como 60000
 - Empleado principiante 50000
- El nombre, código de rango y cédula de los empleados.
- Las diferentes áreas de trabajo dentro de la organización, inicialmente se encuentran unos pocas las cuales se codifican de la siguiente manera:
 - Área de servicios varios, código A01.
 - Área de administración y recursos humanos, código A02
 - Área de producción, código A03
 - Área de calidad, código A04
- Las peticiones de productos de los clientes
- Los productos existentes, cada uno tiene un código diferente asociado.

4.3.2. Solución

- De acuerdo a los conceptos teóricos vistos en clase.
- (6 puntos) ¿Qué estructuras de datos usaría para almacenar los datos de los cargos en la organización? Justifique su elección. ¿Su estructura de datos asegura que se pueda buscar el cargo con menor importancia rápidamente?
- (3 puntos) ¿Qué estructuras de datos usaría para almacenar los datos de los empleados?. Justifique su elección de acuerdo a los tiempos en las operaciones de inserción, búsqueda y borrado de datos
- (6 puntos) ¿Qué estructuras de datos usaría para almacenar los datos de los áreas en la organización? Justifique su respuesta. ¿Como haría usted para minimizar el tiempo de búsqueda de un área específica a partir de su código?.
- (3 puntos) ¿Que estructura de datos usaría para manejar las peticiones de los clientes? Justifique su respuesta.
- (3 puntos) ¿Que estructura de datos usaría para almacenar los productos existentes?.
- (9 puntos) Explique brevemente un algoritmo que permita atender una solicitud de un cliente, proceso que se realiza de esta manera:
 - Obtener la petición más antigua
 - Asignarla a un empleado cualquiera escogido aleatoriamente con rango menor a 75000
 - Almacenar la relación empleado y petición del cliente. Pista: Indique una estructura de datos para almacenar esta relación. Recuerde que pueden existir muchos datos.
 - Generar una respuesta al cliente, suponga que este proceso es automático al almacenar relación empleado y petición del cliente

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 $\epsilon > 0$ y $af(\frac{n}{b}) \leq cf(n)$ entonces $T(n) = \Theta(f(n))$.

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