

## Segundo examen opcional - versión B

### Fundamentos de análisis y diseño de algoritmos

Carlos Andres Delgado S, Ing \*

20 de Diciembre 2017

**Importante:** Se debe escribir el procedimiento realizado en cada punto, con sólo presentar la respuesta, el punto no será válido. La complejidad temporal debe ser dada en términos de  $O(f(n))$ , con la cota  $f(n)$  más ajustada posible.

#### 1. Ordenamiento [50 puntos]

Se desea ordenar un conjunto de números entre 1 y  $n^2$  distribuidos de forma uniforme, es decir que no tenemos números preferidos dentro del conjunto, todos tienen la misma posibilidad de aparecer en el conjunto. Se tienen los siguientes algoritmos de ordenamiento:

##### Algoritmo A

```
para i=0 hasta n-1
    mínimo = i;
    para j=i+1 hasta n
        si arreglo[j] < arreglo[mínimo] entonces
            mínimo = j
        fin si
    fin para
    intercambiar(arreglo[i], arreglo[mínimo])
fin para
```

##### Algoritmo B

```
mientras arreglo no esté ordenado
    reordenar_aleatoriamente(arreglo)
fin mientras
```

##### Algoritmo C

```
algoritmoC(Arreglo, i, j)
    si Arreglo[i] > Arreglo[j] entonces
        intercambiar(Arreglo[i], Arreglo[j])
    fin si

    si (j - i + 1) > 2 entonces
        t = (j - i + 1) / 3
        algoritmoC(Arreglo, i, j-t)
        algoritmoC(Arreglo, i+t, j)
        algoritmoC(Arreglo, i, j-t)
    fin si

    retornar Arreglo

fin

algoritmoC(Arreglo, 1, n)
```

##### Algoritmo D

```
Para i = 2 hasta n hacer
    para j = 0 hasta n-i hacer
        si Arreglo[j] > Arreglo[j+1]
            intercambiar(Arreglo[j], Arreglo[j+1])
        fin para
    fin para
```

1. [10 puntos] Calcule la complejidad del algoritmo A, para mejor, caso promedio y peor caso. Explique el proceso que realiza para calcularla.
2. [10 puntos] Calcule la complejidad del algoritmo B, para mejor, caso promedio y peor caso. Explique el proceso que realiza para calcularla.
3. [10 puntos] Calcule la complejidad del algoritmo C, para mejor, caso promedio y peor caso. Explique el proceso que realiza para calcularla.
4. [10 puntos] Calcule la complejidad del algoritmo D, para mejor, caso promedio y peor caso. Explique el proceso que realiza para calcularla.
5. [10 puntos] Si existe una mejor solución argumente porque. Si no, sustente claramente porque.

#### 2. Programación dinámica y voraz [50 puntos]

En una compañía se está planeando una fiesta. La compañía tiene una estructura jerárquica, donde existen relaciones donde un supervisor es el nodo padre de alguien que supervisa, por lo que el presidente de la compañía esta en la raíz del árbol que representa la estructura de la empresa. Cada empleado tiene una calificación de convivencia, la cual es un número natural. El presidente de la empresa requiere que en la lista de invitados no se encuentren un empleado con su respectivo supervisor. Diseñe un algoritmo que permita maximizar la suma de la convivencia de los invitados de la fiesta cumpliendo los requerimientos del presidente.

**Pista:** La estructura de la compañía es un árbol  $n$ -ario donde el presidente es la raíz.

1. [10 puntos] Indique cómo es la solución ingenua de este problema y su complejidad
2. [10 puntos] Identifique la estrategia divide y vencerás de este problema ¿Como dividimos? ¿Cual es la

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

solución trivial? ¿Cómo combinamos las soluciones de los subproblemas?

3. **[20 puntos]** Especifique la solución dinámica de este problema. Recuerde 1) Caracterizar si es un problema de programación dinámica 2) Especificar la estructura de memorización 3) Plantear cómo se va llenar esta estructura 4) ¿Cómo se calcula la solución a partir de lo realizado en 3?. Muestre cómo se solucionaría un ejemplo con un árbol de altura 3 y al menos 6 nodos. Calcule la complejidad de su solución.
4. **[10 puntos]** Especifique la solución voraz de su problema. Calcule su complejidad