



Segundo examen parcial
FUNDAMENTOS DE ANÁLISIS Y DISEÑO DE ALGORITMOS
Grupo 80 - Duración: 2 horas

Carlos Andres Delgado S, Ing *

18 de Junio de 2014

Nota: Por favor marcar todas las hojas del examen con su **nombre y código de estudiante**. Recuerde colocar los procedimientos debido a que tienen un gran valor en la calificación.

1. Calentamiento [10 puntos]

Se tiene el arreglo A que representa los datos almacenados de un montón.

$$A = [32, 30, 25, 28, 21, 20, 16, 15, 12] \quad (1)$$

Dibuje el montón representado por A e indique si se cumplen las propiedades de orden y forma.

2. Ordenamiento usando Shellsort: Análisis [50 puntos]

Considere el algoritmo ShellSort:

```
ShellSort(A)
   $\delta \leftarrow \text{length}[A]$ 
  while ( $\delta > 1$ )
     $\delta \leftarrow \lceil \frac{\delta}{3} \rceil$ 
     $i \leftarrow 1$ 
    while  $i \leq \delta$ 
      deltaInsertionSort(A, i,  $\delta$ )
       $i \leftarrow i + 1$ 
```

Donde *deltaInsertionSort* es:

```
deltaInsertionSort(A, i,  $\delta$ )
   $j \leftarrow i + \delta$ 
   $n \leftarrow \text{length}[A]$ 
  while  $j \leq n$ 
     $key \leftarrow A[j]$ 
     $k \leftarrow j - \delta$ 
    while ( $k \geq i$  &  $A[k] > key$ )
       $A[k + \delta] \leftarrow A[k]$ 
       $k \leftarrow k - \delta$ 
     $A[k + \delta] \leftarrow key$ 
     $j \leftarrow j + \delta$ 
```

1. [10 puntos] Describa el estado del arreglo A y la variable δ cada vez que el algoritmo *ShellSort* va ejecutar la línea 2, es decir el inicio del ciclo determinado por *while* ($\delta > 1$), si el arreglo es:

$$A = [3, 13, 14, 12, 4, 10, 5, 9, 8, 2, 6, 1, 11, 7] \quad (2)$$

δ	A												
14													

2. [20 puntos] Verifiquemos que ShellSort ordena correctamente
- [6 puntos] Justifique brevemente que *ShellSort* siempre termina.
 - [7 puntos] ¿Cuales son los valores de i y δ en el último llamado a *deltaInsertionSort* en la línea 6?.

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

- c) [7 puntos] ¿Cual es el efecto visible sobre A después de ejecutar el llamado a *deltaInsertionSort*($A,1,1$)?. Justifique su respuesta haciendo referencia al *InsertionSort*
3. [20 puntos] Estudio del algoritmo
- a) [10 puntos] Identifique el peor caso para el algoritmo para ShellSort, es decir el caso donde se hace el mayor número posible de comparaciones. Demuestre con un arreglo de tamaño 9 con números (no repetidos) entre 1 y 9, para el que . ¿Cuántas comparaciones se hacen en su ejemplo?
- b) [10 puntos] Identifique el mejor caso para el algoritmo para ShellSort, es decir el caso donde se hace el menor número posible de comparaciones. Demuestre con un arreglo de tamaño 9 con números (no repetidos) entre 1 y 9, para el que . ¿Cuántas comparaciones se hacen en su ejemplo?
2. [20 puntos] Considere un algoritmo voraz en el cual se plantea una estrategia de almacenar los archivos en orden creciente de tamaños, por ejemplo para el caso del punto anterior se almacenarían en este orden: 35,42 y 60 Mb. ¿Esta estrategia voraz es correcta y optima?. Demuestre con un contraejemplo.

3. Programación dinámica y programación voraz [40 puntos]

3.1. Conceptos teóricos [10 puntos]

Indique si es verdadero o falso. Justifique su respuesta.

1. [10 puntos] La programación voraz garantiza que siempre se encuentra la solución optima.

3.2. Análisis de algoritmos [30 puntos]

Considere n archivos de tamaños $\{m_1, m_2, \dots, m_n\}$. El problema del almacenamiento optimo de cinta consiste en encontrar el mejor orden para almacenar los archivos en la cinta de manera que la lectura de los mismos sea la menos costosa. Debe tenerse en cuenta

- La lectura de cada archivo comienza con la cinta completamente devuelta.
- Cada lectura en un archivo toma un tiempo igual a la longitud de los archivos precedentes
- Los archivos son leídos en orden inverso al que son almacenados.

1. [10 puntos] Si se tienen tres archivos que miden 35, 60 y 42 Mb. ¿Cual es la mejor manera de almacenarlos? ¿Cual es el costo de leer cada archivo?.