



I EXAMEN PARCIAL

Viernes 8 de abril

El examen consta de 6 preguntas que suman 117, pero no se reconocerán más de 110 (10% extra). Cada pregunta empieza con un indicador de su puntaje y del tema tratado. Si la pregunta tiene subítemes, el puntaje de cada uno de ellos es indicado dentro de los subítemes. Se recomienda echar un vistazo a los temas de las preguntas y a su puntaje antes de resolver el examen, para así distribuir su tiempo y esfuerzo de la mejor manera. El examen se puede realizar con lápiz o lapicero. Las preguntas pueden ser respondidas en cualquier orden pero se debe indicar claramente el número de cada pregunta. **No se permite el uso de calculadora.**

1. [12 pts.] *Matemática básica.*

Demuestre que:

$$1 + r + r^2 + \dots + r^n = \frac{1 - r^{n+1}}{1 - r}.$$

2. [32 pts.] *Solución de recurrencias.*

Resuelva las siguientes recurrencias asumiendo que $T(n) = \Theta(1)$ para $n \leq 1$, que $c > 0$ y que $k \geq 1$.

a) $T(n) = T(n - k) + c$. [7 pts.]

b) $T(n) = 2T(n/\sqrt{2}) + n$. [7 pts.]

c) $T(n) = T(n/2) + (\lg n)^2$. [18 pts.]

Observación. $1 + 2^2 + 3^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$

3. [19 pts.] *Análisis de la correctitud y del tiempo de ejecución de un algoritmo iterativo.*

El siguiente algoritmo calcula el factorial de un número:

```
factorial(n)
  si n = 0
    devuelva 0
  f = 1
  para i = 1..n
    f = f * i
  devuelva f
```

- a) Demuestre que el algoritmo es correcto, identificando el invariante del ciclo [5 pts.] y los pasos de inicialización [3 pts.], mantenimiento [5 pts.] y terminación [3 pts.].
 - b) Determine el comportamiento asintótico del tiempo de ejecución del algoritmo. [3 pts.]
4. [17 pts.] *Análisis del tiempo de ejecución de un algoritmo recursivo.*
 El siguiente algoritmo implementa recursivamente la búsqueda binaria de un elemento x en el subarreglo **ordenado** $A[i..j]$:

```

BusquedaBinaria(x, A, i, j)
    si j < i
        devuelva -1 // no se encontró
    m = (i + j) / 2 // punto medio
    si A[m] > x
        devuelva BusquedaBinaria(x, A, i, m-1)
    sino, si A[m] < x
        devuelva BusquedaBinaria(x, A, m+1, j)
    sino
        devuelva m // se encontró (en posición m)
  
```

- a) Escriba una fórmula recursiva para el tiempo de ejecución del algoritmo en el peor caso. [12 pts.]
 - b) Determine el comportamiento asintótico del tiempo de ejecución en el peor caso. [5 pts.]
5. [35 pts.] *Simulación de la ejecución de algoritmos.*
 Simule la ejecución del ordenamiento *alfabético* ascendente del arreglo $A = \langle do, re, mi, fa, so \rangle$, utilizando los siguientes algoritmos y mostrando el (los) estado(s) del (los) (sub)arreglos después de cada «paso» (iteración del ciclo principal o llamado a [sub]función del algoritmo) Después del primer paso incorrecto el resto de pasos no suman puntos. [5 pts. c/ simulación, excepto ordenamiento por montículos: 5 pts. por monticularizar y 5 pts. por ordenar].

Observación. *Se enfatiza que la última palabra en el arreglo es so y no sol, y que el ordenamiento deseado es el alfabético (do, fa, mi, re, so) y no el de la escala musical (do, re, mi, fa, so).*

- a) Ordenamiento por selección.
- b) Ordenamiento por inserción.
- c) Ordenamiento por mezcla [*mergesort*]. (Si el tamaño del [sub]arreglo es impar, *redondee* el punto medio hacia abajo, es decir, que la parte izquierda del subarreglo sea más pequeña que su parte derecha, y no al revés. Por ejemplo, un arreglo de tamaño 5 debería ser partido en un arreglo de tamaño 2 [a la izquierda] y uno de tamaño 3 [a la derecha]).

- d) Ordenamiento rápido [*quicksort*] (usando el primer o último elemento como pivote, **indique cuál**).
- e) Ordenamiento usando montículos (*heapsort*)
- f) Ordenamiento por residuos [*radixsort*] (Asuma que en cada pasada se ordena un caracter distinto).

6. [2 pts.] *Educación vial*

- a) Si una calle *no cuenta* con aceras, ¿por qué lado de la calle deben caminar los peatones? ¿Por qué? [1 pto.]
- b) Si una calle *cuenta* con aceras, ¿por dónde deben caminar los peatones? ¿Por qué? [1 pto.]