



CI-1221 Estructuras de Datos y Análisis de Algoritmos
I ciclo 2013, grupo 2

I EXAMEN PARCIAL

Martes 30 de abril

Apellido(s): _____ Nombre: _____ Carné: _____

El examen consta de 5 preguntas que suman 131 puntos, pero no se reconocerán más de 110 (10 % extra). Cada pregunta indica el tema tratado y su valor. 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.

Las preguntas se pueden responder en cualquier orden, pero se debe indicar en la tabla mostrada abajo los números de página en la que están las respuestas. Para ello numere las hojas del cuaderno de examen en la esquina superior externa de cada página.

El examen se puede realizar con lápiz o lapicero. *No se permite el uso de dispositivos electrónicos (calculadoras, teléfonos, audífonos, etc.).*

Las preguntas del examen están inspiradas en la próxima visita del presidente de EE. UU., Barack Obama, a Costa Rica.

Pregunta	Puntos	Páginas	Calificación
1. <i>Algoritmos recursivos</i>	15		
2. <i>Algoritmos iterativos</i>	30		
3. <i>Solución de recurrencias</i>	20		
4. <i>Simulación de ejecución de algoritmos</i>	44		
5. <i>Algoritmos de ordenamiento iterativos</i>	22		
Total	131		

1. *Algoritmos recursivos.* [15 pts.]

Durante su visita a Costa Rica, el presidente Obama desea mantener contacto regular con su esposa Michelle. Para ello ha solicitado a sus colaboradores que incluyan dentro de su agenda momentos para llamarla. El algoritmo *recursivo* propuesto fue el siguiente:

Si hay que realizar una actividad:

- Realizar la actividad.

Si hay que realizar $n > 1$ actividades:

- Realizar recursivamente $\lfloor n/2 \rfloor$ actividades.
 - Llamar a Michelle.
 - Realizar recursivamente las $\lceil n/2 \rceil$ actividades restantes.
- a) Asumiendo que las actividades toman un tiempo t_a (el mismo para todas las actividades) y que las llamadas toman un tiempo t_l (el mismo para todas las llamadas), determine una fórmula recursiva para el tiempo que toma realizar n actividades y hacer las llamadas telefónicas correspondientes [5 pts.] y resuelva la recurrencia [5 pts.]
- b) Calcule la cantidad de llamadas que permite al presidente Obama hacer este algoritmo en un día con n actividades. [5 pts.]

2. *Algoritmos iterativos.* [30 pts.]

- a) Escriba un algoritmo iterativo que produzca el mismo patrón de actividades y llamadas que el algoritmo recursivo de la pregunta 1. [10 pts.]
- b) Establezca la correctitud del algoritmo mostrando que permite ejecutar las n actividades planeadas y realizar el número de llamadas telefónicas calculado en el punto b) de la pregunta 1. Para ello identifique el invariante del ciclo [5 pts.] y muestre cómo los pasos de inicialización [5 pts.], mantenimiento [5 pts.] y terminación [5 pts.] permiten determinar la correctitud del algoritmo.
- c) Escriba una fórmula para el tiempo de ejecución del algoritmo [5 pts.] y determine una cota asintótica lo más ajustada posible [5 pts.].

3. *Solución de recurrencias.* [20 pts.]

Resuelva la siguiente recurrencia:

$$T(n) = \begin{cases} T\left(\frac{n}{2} - 1\right) + c, & n \geq 2; \\ \Theta(1), & n < 2. \end{cases}$$

4. *Simulación de ejecución de algoritmos.* [44 pts.]

Simule la ejecución de los algoritmos especificados abajo sobre el arreglo $A = \langle cj, rr, bghw, cb, bgw, ob \rangle$, correspondiente a las siglas de los últimos seis presidentes de los EE. UU. con el apellido antepuesto a los nombres (ver cuadro 1). Muestre el (los) estado(s) del (los) (sub)arreglos después de cada paso principal (iteración del ciclo principal o llamado a subrutina del algoritmo). Si en un paso no muestra el estado de algunas entradas del arreglo, se asume que estas conservan el estado anterior. (Se sugiere hacer uso de esto para ahorrar tiempo). El criterio de ordenamiento es el *alfabético*, con prioridad del apellido sobre los nombres. *Asuma que el carácter vacío es menor que cualquier carácter no vacío.* Después del primer paso incorrecto el resto de pasos no suman puntos.

Cuadro 1: Últimos seis presidentes de EE. UU.

Nombre	Abreviatura	Periodo del mandato
Carter, Jimmy	<i>cj</i>	1977–1981
Reagan, Ronald	<i>rr</i>	1981–1989
Bush, George H. W.	<i>bghw</i>	1989–1993
Clinton, Bill	<i>cb</i>	1993–2001
Bush, George W.	<i>bgw</i>	2001–2009
Obama, Barack	<i>ob</i>	2009–presente

a) Ordenamiento por selección. [5 pts.]

	Entrada					
	1	2	3	4	5	6
Estado inicial	<i>cj</i>	<i>rr</i>	<i>bghw</i>	<i>cb</i>	<i>bgw</i>	<i>ob</i>
Iteración 1						
Iteración 2						
Iteración 3						
Iteración 4						
Iteración 5						

b) Ordenamiento por inserción. [5 pts.]

	Entrada					
	1	2	3	4	5	6
Estado inicial	<i>cj</i>	<i>rr</i>	<i>bghw</i>	<i>cb</i>	<i>bgw</i>	<i>ob</i>
Iteración 1						
Iteración 2						
Iteración 3						
Iteración 4						
Iteración 5						

c) Ordenamiento por mezcla (*mergesort*). (Si el tamaño del [sub]arreglo es impar, *redondee* el punto medio hacia abajo, es decir, la parte izquierda del subarreglo debe ser más pequeña que la parte derecha. Por ejemplo, un arreglo de tamaño 3 debe ser partido en un arreglo de tamaño 1 [a la izquierda] y uno de tamaño 2 [a la derecha]). [5 pts.]

d) Ordenamiento usando montículos (*heapsort*). [3 pts. monticularizar; 5 pts. ordenar]

e) Ordenamiento rápido (*quicksort*). Usar como pivote el último elemento del (sub) arreglo. [5 pts.]

f) Ordenamiento por residuos (*radixsort*). Muestre lo siguiente en cada llamado a la subrutina de ordenamiento por conteo:

i. El histograma *C*. [4 pts.]

ii. El histograma acumulativo *C*. [4 pts.]

iii. El arreglo resultante *B*. [4 pts.]

iv. El estado final del histograma acumulativo *C* (después de producir el arreglo *B*). [4 pts.]

5. Algoritmos de ordenamiento iterativos. [22 pts.]

En una ocasión, el director ejecutivo de Google le preguntó a Barack Obama, simulando una

entrevista de trabajo, cuál era la forma más eficiente de ordenar un millón de enteros de 32 bits, a lo que el Sr. Obama respondió «*I think Bubble Sort would be the wrong way to go*».

He aquí el algoritmo de ordenamiento de burbuja en el lenguaje de programación C:

```
1 void bubbleSort(int A[], int n) {
2     int i, j, temp;
3     for (i = n - 1; i > 0; i--) {
4         for (j = 1; j <= i; j++) {
5             if (A[j-1] > A[j]) {
6                 temp = A[j-1];
7                 A[j-1] = A[j];
8                 A[j] = temp;
9             }
10        }
11    }
12 }
```

Asumiendo que una asignación toma tiempo t_a , una suma o resta t_s , una comparación t_c , un incremento o decremento t_i :

- Escriba una fórmula para el tiempo de ejecución $T(n)$ del algoritmo. [5 pts.]
- Calcule una cota lo más ajustada posible para $T(n)$ en el peor caso. [5 pts.]
- Indique si el Sr. Obama tenía o no razón y por qué. [2 pts.]