

CI-1221 Estructuras de Datos y Análisis de Algoritmos
I ciclo de 2018

I EXAMEN PARCIAL

Viernes 4 de mayo, 5:00 p. m. – 7:30 p. m.

Nombre: _____ Carné: _____

El examen consta de 9 preguntas que suman 148 puntos, pero no se reconocerán más de 110 (10 % extra). Las preguntas se pueden responder en cualquier orden, pero se deben indicar en el cuadro que está abajo los números de página del cuaderno de examen en la que están las respuestas. Para esto debe numerar las hojas del cuaderno de examen en la esquina superior externa de cada página. Si la respuesta está en el enunciado del examen favor indicarlo con la letra 'E' en vez del número de página. El examen se puede realizar con lápiz o lapicero. *No se permite el uso de dispositivos electrónicos: calculadoras, teléfonos, audífonos, etc.*

Pregunta	Puntos	Páginas o 'E'	Calificación
1. <i>Algoritmos recursivos</i>	25		
2. <i>Algoritmos iterativos</i>	42		
3. <i>Solución de recurrencias</i>	22		
4. <i>Ordenamiento por selección</i>	5		
5. <i>Ordenamiento por inserción</i>	5		
6. <i>Ordenamiento por mezcla</i>	4		
7. <i>Ordenamiento por montículos</i>	9		
8. <i>Ordenamiento rápido</i>	11		
9. <i>Ordenamiento por residuos</i>	25		
Total	148		

Parte I

Una *fracción continua* finita permite expresar un número racional $x = p/q$ como una secuencia de números enteros $[a_0; a_1, a_2, \dots, a_n]$, usualmente corta, tal que

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{\ddots + \frac{1}{a_n}}}} \quad (1)$$

Por ejemplo, la fracción 415/93 se puede expresar mediante la secuencia $[4; 2, 6, 7]$. (Los números irracionales se pueden expresar con secuencias infinitas que, aun truncadas, producen muy buenas aproximaciones; por ejemplo $[3; 7, 16]$ produce 3.14159292..., una muy buena aproximación de π .) También se usa la siguiente útil notación introducida por C. F. Gauss:

$$x = a_0 + \mathop{\text{K}}_{k=1}^n \frac{1}{a_k}, \quad (2)$$

donde la letra K representa el término en alemán *kettenbruch*, que significa «fracción continua».

1. *Algoritmos recursivos.* [25 pts.]

- a) Escriba un algoritmo recursivo en C o C++ que devuelva la representación en punto flotante de una secuencia $A = [a_0; a_1, a_2, \dots, a_n]$. Use el siguiente encabezado:

```
float fracContRec(int A[], int n);
```

Puede usar métodos auxiliares si lo necesita. [15 pts.]

- b) Escriba una fórmula recursiva para el tiempo de ejecución del algoritmo. Asuma que el tiempo de ejecución de una asignación es t_a , el de una comparación es t_c , el de un incremento (operador $++$) es t_i , el de una suma o resta es t_s y el de una multiplicación o división es t_m . [5 pts.] (Esta pregunta sumará puntos solo si el algoritmo es correcto).
- c) Determine una cota asintótica del tiempo de ejecución del algoritmo lo más ajustada posible. [5 pts.] (Esta pregunta sumará puntos solo si el algoritmo y la fórmula recursiva son correctos).

2. *Algoritmos iterativos.* [42 pts.]

- a) Escriba una versión iterativa, `fracContIt`, del algoritmo de la pregunta 1. [15 pts.]
- b) Establezca la correctitud del algoritmo mostrando que calcula correctamente la ecuación 1. Para ello identifique el invariante del ciclo (el más externo, si hay ciclos anidados) [5 pts.] y muestre cómo los pasos de inicialización [5 pts.], mantenimiento [5 pts.] y terminación [2 pts.] implican la correctitud del algoritmo.
- c) Escriba una fórmula para el tiempo de ejecución del algoritmo asumiendo que el tiempo de ejecución de una asignación es t_a , el de una comparación es t_c , el de un incremento (operador $++$) es t_i , el de una suma o resta es t_s y el de una multiplicación o división es t_m . [5 pts.] (Esta pregunta sumará puntos solo si el algoritmo es correcto).
- d) Determine una cota asintótica del tiempo de ejecución del algoritmo lo más ajustada posible. [5 pts.] (Esta pregunta sumará puntos solo si el algoritmo y la fórmula son correctos).

3. *Solución de recurrencias.* [22 pts.]

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

- a) $T(n) = 2T(3n/4) + c(n \log n)^2$. [7 pts.]
- b) $T(n) = T((n-1)/2) + c$. [15 pts.]

Parte II

Esta parte del examen pide ordenar los diputados y diputadas del periodo 2018-2022 por las iniciales de su primer nombre y primer apellido, usando distintos algoritmos de ordenamiento. Sus nombres, fotografías y cargos se muestra en las figuras 1 a 7, tomadas del periódico La Nación en línea. Asuma que al inicio los(as) diputados(as) están colocados en el arreglo en orden de renglón, es decir, de izquierda a derecha, avanzando al siguiente renglón al acabar uno. (Equivalentemente, asuma que están ordenados primero por orden alfabético de partido y luego por lugar dentro del partido: 1º, 2º, 3º, etc.).

4. *Ordenamiento por selección.* [5 pts.]

Simule la ejecución del algoritmo de ordenamiento por selección para ordenar las iniciales de los(as) diputados(as) de Limón. Escriba en la iteración «0» el estado inicial del arreglo y muestre el estado del arreglo al finalizar cada una de las iteraciones siguientes del ciclo principal (externo). Si no muestra el estado de una casilla se asume que conserva el valor que tenía en la iteración anterior. (Después de la primer inserción incorrecta el resto de inserciones no suman puntos).

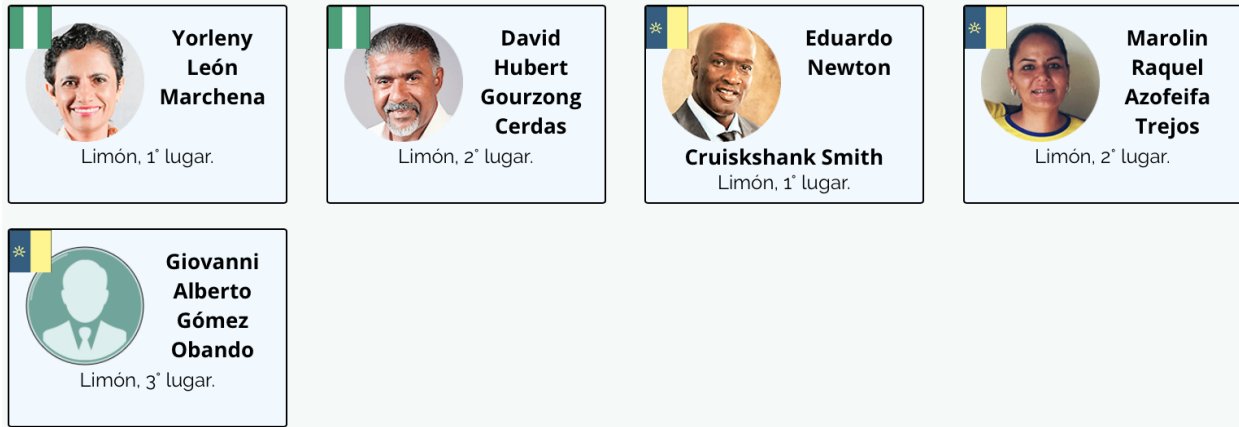


Figura 1: Diputados(as) de Limón.

	Posición				
It.	1	2	3	4	5
0					
1					
2					
3					
4					

5. *Ordenamiento por inserción.* [5 pts.]

Simule la ejecución del algoritmo de ordenamiento por inserción para ordenar las iniciales de los(as) diputados(as) de Puntarenas. Escriba en la iteración «1» el estado inicial del arreglo y muestre el estado del arreglo al finalizar cada una de las iteraciones siguientes del ciclo principal (externo). Si no muestra el estado de una casilla se asume que conserva el valor que tenía en la iteración anterior. (Después de la primer inserción incorrecta el resto de inserciones no suman puntos).

	Posición				
It.	1	2	3	4	5
1					
2					
3					
4					
5					

6. *Ordenamiento por mezcla.* [4 pts.]

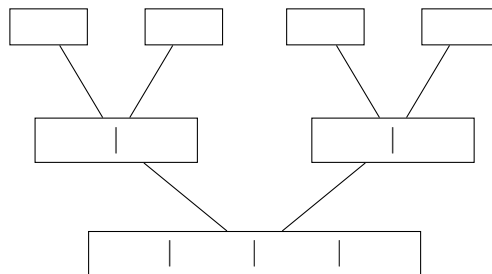
Simule la ejecución del algoritmo de ordenamiento por mezcla para ordenar las iniciales de los(as) diputados(as) de Guanacaste. Muestre en la parte superior el estado inicial del arreglo [1 pto.] y luego el estado de cada uno de los subarreglos al finalizar cada una de las mezclas [1 pto. c/ mezcla]. (Después de la primer mezcla incorrecta el resto de mezclas no suman puntos).



Figura 2: Diputados(as) de Puntarenas.



Figura 3: Diputados(as) de Guanacaste.



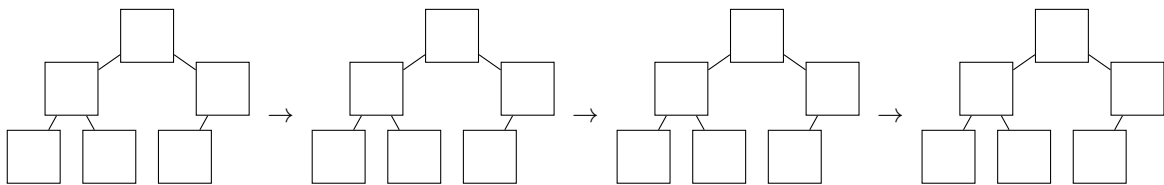
7. *Ordenamiento por montículos.* [9 pts.]

Simule la ejecución del algoritmo de ordenamiento por montículos para ordenar las iniciales de los(as) diputados(as) de Heredia. Muestre el estado inicial del montículo y después de cada llamado a CORREGIR-CIMA y EXTRAER-MÁXIMO. Si deja un nodo vacío, se asume que tiene el mismo valor que en el paso anterior. (Los nodos con línea discontinua los puede dejar vacíos si lo desea, puesto que su contenido es trivial). (Después de la primer operación incorrecta el resto de operaciones no suman puntos).

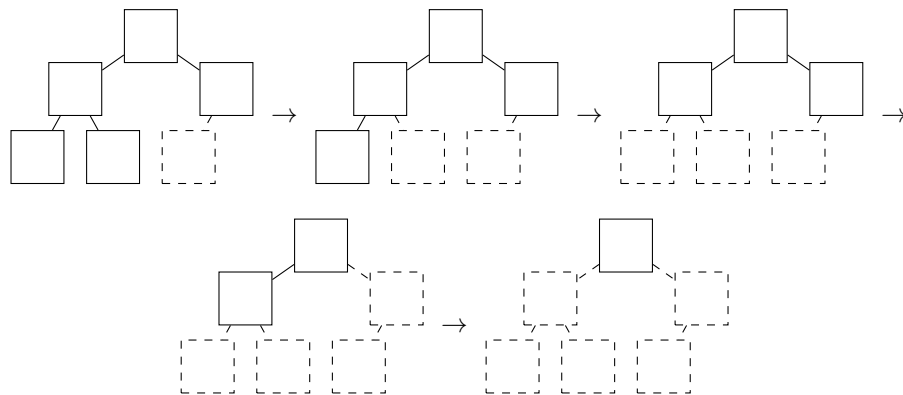


Figura 4: Diputados(as) de Heredia.

MONTICULARIZAR:



ORDENAR:



8. Ordenamiento rápido. [11 pts.]

Simule la ejecución del algoritmo de ordenamiento rápido para ordenar las iniciales de los(as) diputados(as) de Alajuela. Muestre en el primer renglón el estado inicial del arreglo y luego el estado después de cada llamado al método PARTICIÓN, e indique la posición del pivote devuelto por él. Si no muestra el estado de una casilla se asume que conserva el valor que tenía en la iteración anterior. (Después del primer estado incorrecto el resto de estados no suman puntos).



Figura 5: Diputados(as) de Alajuela.

Llamado	Estado del arreglo											Posición del pivote
	1	2	3	4	5	6	7	8	9	10	11	
1.º	1	2	3	4	5	6	7	8	9	10	11	
2.º	1	2	3	4	5	6	7	8	9	10	11	
3.º	1	2	3	4	5	6	7	8	9	10	11	
4.º	1	2	3	4	5	6	7	8	9	10	11	
5.º	1	2	3	4	5	6	7	8	9	10	11	
6.º	1	2	3	4	5	6	7	8	9	10	11	
7.º	1	2	3	4	5	6	7	8	9	10	11	
8.º	1	2	3	4	5	6	7	8	9	10	11	
9.º	1	2	3	4	5	6	7	8	9	10	11	
10.º	1	2	3	4	5	6	7	8	9	10	11	

9. *Ordenamiento por residuos.* [25 pts.]

Simule la ejecución del algoritmo de ordenamiento por residuos para ordenar las iniciales de los(as) diputados(as) de San José. Muestre en el cuadro 1 lo siguiente para cada llamado a la subrutina de ordenamiento por conteo: el histograma C [5 pts.], el histograma acumulativo C' [5 pts.], el arreglo resultante B [10 pts.] y el estado final del histograma acumulativo C'' (después de producir el arreglo B) [5 pts.]. (Después del primer arreglo incorrecto el resto no suman puntos).














 Nielsen Pérez Pérez San José, 1° lugar.	 Víctor Morales Mora San José, 2° lugar.	 Paola Vega Rodríguez San José, 3° lugar.	 Enrique Sánchez Carballo San José, 4° lugar.
 José María Villalta Florez-Estrada San José, 1° lugar.	 Wálter Muñoz Céspedes San José, 1° lugar.	 Zoila Rosa Volio Pacheco San José, 2° lugar.	 Carlos Ricardo Benavides Jiménez San José, 1° lugar.
 Silvia Hernández Sánchez San José, 2° lugar.	 Wagner Jiménez Zúñiga San José, 3° lugar.	 Ana Karine Niño Gutiérrez San José, 4° lugar.	 Otto Roberto Vargas Víquez San José, 1° lugar.
 Carlos Luis Avendaño Calvo San José, 1° lugar.	 Floria María Segreda Sagot San José, 2° lugar.	 Harllan Hoepelman Páez San José, 3° lugar.	 Ivonne Acuña Cabrera San José, 4° lugar.
 María Vita Monge Granados San José, 1° lugar.	 Pedro Muñoz Fonseca San José, 2° lugar.	 Shirley Díaz Mejía San José, 3° lugar.	

Figura 6: Diputados(as) de San José.

