

CI-0116 Análisis de Algoritmos y Estructuras de Datos
II ciclo de 2019

I EXAMEN PARCIAL

Jueves 26 de setiembre

Nombre: _____ Carné: _____

El examen consta de 10 preguntas que suman 150 puntos, pero no se reconocerán más de 110 (10 % extra). Las preguntas se pueden responder en cualquier orden, pero se debe indicar en el cuadro de 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	Calificación
1. <i>Algoritmos recursivos</i>	20		
2. <i>Algoritmos iterativos</i>	35		
3. <i>Solución de recurrencias</i>	20		
4. <i>Ordenamiento por selección</i>	9		
5. <i>Ordenamiento por inserción</i>	9		
6. <i>Ordenamiento por mezcla</i>	9		
7. <i>Ordenamiento por montículos</i>	14		
8. <i>Ordenamiento rápido</i>	9		
9. <i>Ordenamiento por residuos</i>	16		
10. <i>Búsqueda de la mediana</i> (opcional)	9		
Total	150		

Parte I

1. Algoritmos recursivos. [20 pts.]

- a) Implemente una función recursiva en C++ que halle en un arreglo de números $A = \langle a_0, a_1, \dots, a_{n-1} \rangle$ un subarreglo de suma máxima (SSM), es decir, un arreglo $B = \langle a_i, a_{i+1}, \dots, a_j \rangle$ ($0 \leq i \leq j \leq n$) tal que la suma de los números en B sea lo más grande posible. [10 pts.]. (Precaución: A puede contener números negativos). Use el siguiente encabezado:

```
struct par SSM(int * A, int n);
```

donde el subarreglo B queda definido por los valores de la estructura *par* definida así:

```
struct pair {  
    int i;  
    int j;  
}
```

Puede usar métodos auxiliares si lo necesita.

- b) Escriba una fórmula recursiva para el tiempo de ejecución $T(n)$ del algoritmo. [5 pts., válidos solo si la parte a) es correcta].
c) Determine una cota asintótica para el tiempo de ejecución del algoritmo. [5 pts., válidos solo si las partes a) y b) son correctas].

2. Algoritmos iterativos. [35 pts.]

- a) Escriba una versión iterativa del algoritmo de la pregunta 1. [10 pts.]. Use el siguiente encabezado:

```
struct par SSMittr(int * A, int n);
```

- b) Establezca la correctitud del algoritmo. Para ello identifique el invariante del ciclo (el más externo, si hay ciclos anidados) [5 pts.] y muestre cómo los pasos de inicialización [3 pts.], mantenimiento [5 pts.] y terminación [2 pts.] implican la correctitud del algoritmo.
c) Escriba una fórmula para el tiempo de ejecución del algoritmo $T(n, m)$ [5 pts.] y determine una cota asintótica lo más ajustada posible [5 pts.]. [Válido solo si la parte a) es correcta].

3. Solución de recurrencias. [20 pts.]

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

- a) $T(n) = 2T(n/2) + c \log n$. [5 pts.]
b) $T(n) = T(n/2) + c \log(n^2)$. [15 pts.]

Cuadro 1: Resumen de los ejes y metas del *Plan de descarbonización* (2018-2050). (Fuente: presidencia.go.cr/comunicados/2019/02/sintesis-plan-nacional-de-descarbonizacion-2018-2050).

Eje	Metas
Buses, taxis y trenes	Desarrollar un sistema de transporte público y movilidad compartida seguro y alimentado por energías limpias.
Carros y motos	Transformar progresivamente la flota vehicular a vehículos sin emisiones y promover los autos compartidos autónomos.
Camiones	Reducir el impacto ambiental del transporte de carga en el país favoreciendo tecnologías con eficiencia energética y vehículos con baja emisión de carbono.
Electricidad	Consolidar un sistema eléctrico capaz de abastecer y gestionar energía renovable a un bajo costo. Mejorar la gestión mediante la digitalización de los procesos institucionales y comerciales.
Edificios	Desarrollar edificaciones de uso comercial, residencial e institucional bajo estándares de alta eficiencia y bajas emisiones.
Industria	Modernizar el sector industrial mediante la aplicación de procesos eléctricos sostenibles más eficientes.
Residuos	Desarrollar un sistema de gestión integrada de residuos basado en una separación, reutilización, revalorización y disposición final de máxima eficiencia y con bajas emisiones.
Alimentos	Adoptar tecnologías alimentarias eficiente y bajas en emisiones, tanto para el consumo local como para la exportación
Ganado	Consolidar modelos ganaderos basados en la eficiencia productiva y la disminución de gases de efecto invernadero.
Biodiversidad	Consolidar un modelo de gestión de territorios rurales, urbanos y costeros que facilite la protección de la biodiversidad, el incremento y el mantenimiento de la cobertura forestal y servicios ecosistémicos a partir de soluciones basadas en la naturaleza.

Parte II

Hoy, 26 de setiembre, Costa Rica recibe en Nueva York el galardón de *Campeones de la Tierra 2019* de la Organización de las Naciones Unidas (ONU), en la categoría de liderazgo político. Se le da al país por sus esfuerzos durante décadas en pro de la descarbonización y por su reciente propuesta de trabajo *Plan nacional de descarbonización*, resumido en el cuadro 1.

4. *Ordenamiento por selección.* [9 pts.]

Simule la ejecución del algoritmo de ordenamiento por selección sobre el arreglo que aparece a continuación. Muestre el estado del arreglo al finalizar cada una de las iteraciones 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 no suman puntos.

	Posición									
Itn.	1	2	3	4	5	6	7	8	9	10
	Buses	Carros	Camiones	Electricidad	Edificios	Industria	Residuos	Alimentos	Ganado	Biodiversidad
1										
2										
3										
4										
5										
6										
7										
8										
9										

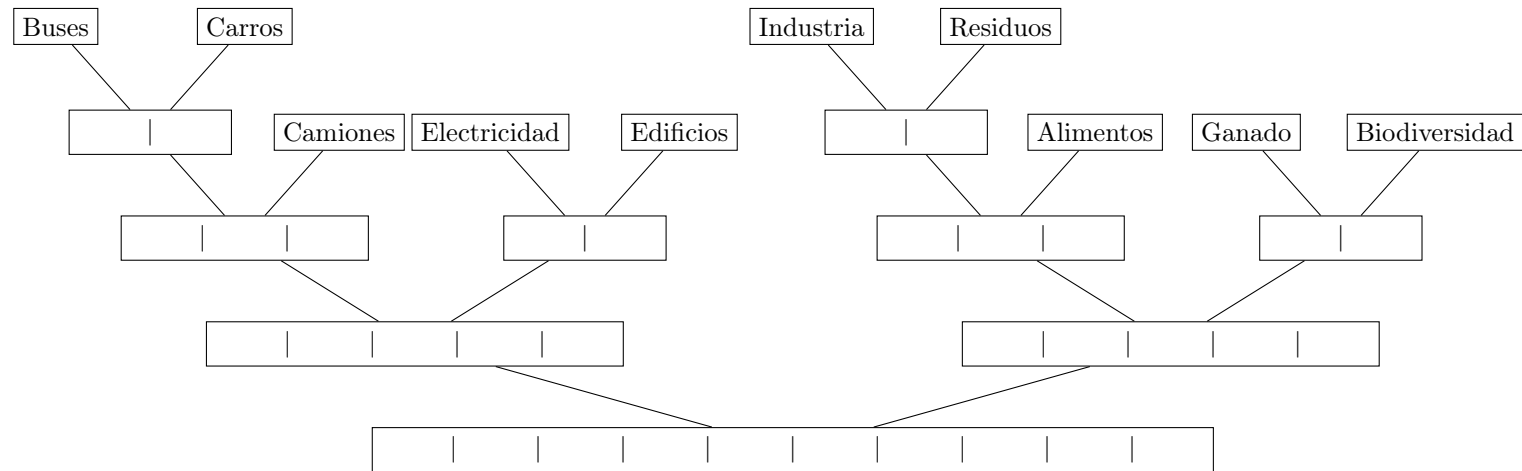
5. *Ordenamiento por inserción.* [9 pts.]

Simule la ejecución del algoritmo de ordenamiento por inserción sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada una de las iteraciones 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 no suman puntos.

	Posición									
Itn.	1	2	3	4	5	6	7	8	9	10
	Buses	Carros	Camiones	Electricidad	Edificios	Industria	Residuos	Alimentos	Ganado	Biodiversidad
2										
3										
4										
5										
6										
7										
8										
9										
10										

6. *Ordenamiento por mezcla.* [9 pts.]

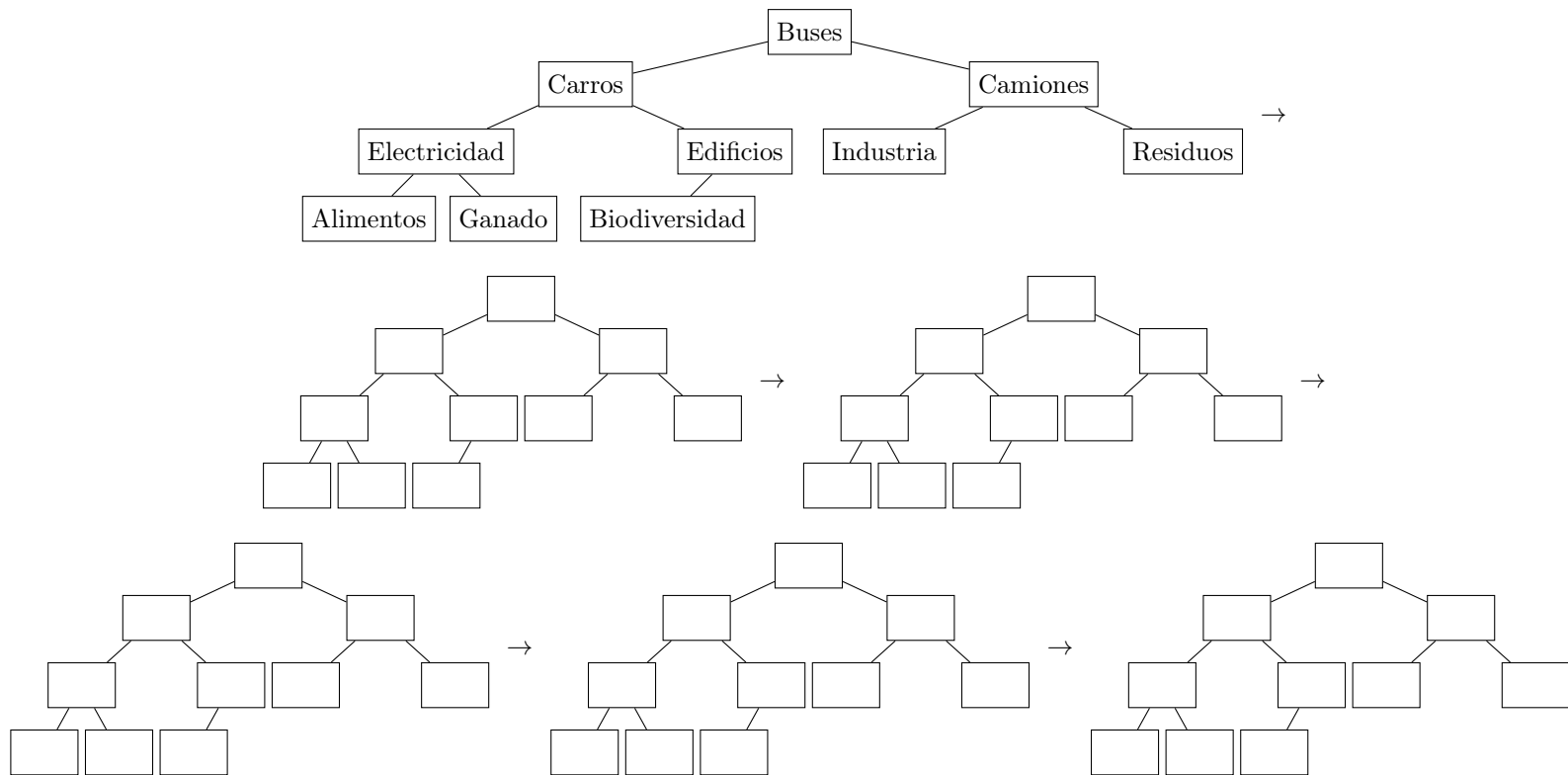
Simule la ejecución del algoritmo de ordenamiento por mezcla sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada una de las mezclas. Si no muestra el estado de una casilla se asume que conserva el valor que tenía en la iteración anterior. Después de la primer mezcla incorrecta el resto de mezclas no suman puntos.



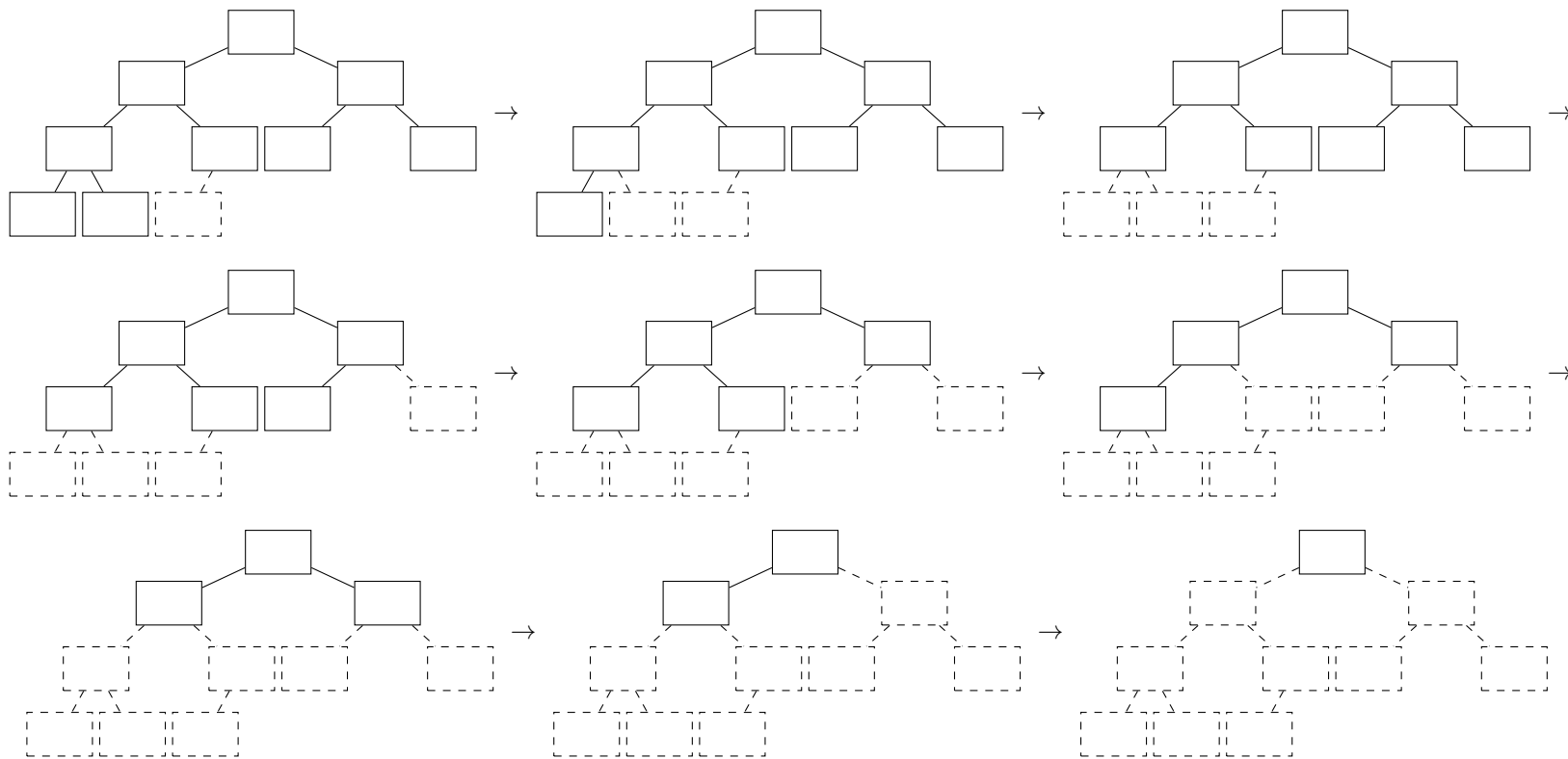
7. Ordenamiento por montículos. [14 pts.]

Simule la ejecución del algoritmo de ordenamiento por montículos sobre el arreglo mostrado abajo en forma de árbol. Muestre el estado del *montículo* después de cada llamado a CORREGIR-CIMA en la monticularización y EXTRAER-MÁXIMO en el ordenamiento. 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 no suman puntos.

MONTICULARIZAR:



ORDENAR:



8. *Ordenamiento rápido*. [9 pts.]

Simule la ejecución del algoritmo de ordenamiento rápido sobre el arreglo mostrado abajo. Muestre el estado del arreglo después de cada llamado a PARTICIÓN e indique la posición del pivote devuelto. 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 no suman puntos.

Llamado	Estado del arreglo										Posición del pivote
	1	2	3	4	5	6	7	8	9	10	
	Buses	Carros	Camiones	Electricidad	Edificios	Industria	Residuos	Alimentos	Ganado	Biodiversidad	
1.º	1	2	3	4	5	6	7	8	9	10	
2.º	1	2	3	4	5	6	7	8	9	10	
3.º	1	2	3	4	5	6	7	8	9	10	
4.º	1	2	3	4	5	6	7	8	9	10	
5.º	1	2	3	4	5	6	7	8	9	10	
6.º	1	2	3	4	5	6	7	8	9	10	
7.º	1	2	3	4	5	6	7	8	9	10	
8.º	1	2	3	4	5	6	7	8	9	10	
9.º	1	2	3	4	5	6	7	8	9	10	

9. *Ordenamiento por residuos*. [16 pts.]

Simule la ejecución del algoritmo de ordenamiento por residuos sobre el arreglo A de números en *base hexal* mostrado abajo. Muestre lo siguiente para cada llamado a la subrutina de ordenamiento por conteo: el histograma C [2 pts.], el histograma acumulativo C' [2 pts.], el arreglo resultante B [8 pts.] y el estado final del histograma acumulativo C'' (después de producir el arreglo B) [4 pts.]. Después del primer arreglo incorrecto el resto de arreglos no suman puntos.

A		C :	C' :	C'' :	$B \rightarrow A$	C :	C' :	C'' :	B
			0 1 2 3 4 5			0 1 2 3 4 5		0 1 2 3 4 5	
1	43				1				1
2	14				2				2
3	50				3				3
4	02				4				4
5	21				5				5
6	33				6				6
7	42				7				7
8	03				8				8

10. *Búsqueda de la mediana* (opcional). [9 pts.]

Simule la ejecución del algoritmo de SELECCIÓN para encontrar la mediana (quinto elemento más pequeño) sobre el arreglo mostrado abajo. Muestre el estado del arreglo después de cada llamado a PARTICIÓN e indique la posición del pivote devuelto. Si no muestra el estado de una casilla se asume que conserva el valor que tenía en la iteración anterior. Asuma que la subrutina PARTICIÓN usa el último elemento del subarreglo como pivote. Después del primer estado incorrecto el resto de estados no suman puntos.

Llamado	Estado del arreglo										Posición del pivote
	1	2	3	4	5	6	7	8	9	10	
	Buses	Carros	Camiones	Electricidad	Edificios	Industria	Residuos	Alimentos	Ganado	Biodiversidad	
1.º	1	2	3	4	5	6	7	8	9	10	
2.º	1	2	3	4	5	6	7	8	9	10	
3.º	1	2	3	4	5	6	7	8	9	10	
4.º	1	2	3	4	5	6	7	8	9	10	
5.º	1	2	3	4	5	6	7	8	9	10	
6.º	1	2	3	4	5	6	7	8	9	10	
7.º	1	2	3	4	5	6	7	8	9	10	
8.º	1	2	3	4	5	6	7	8	9	10	
9.º	1	2	3	4	5	6	7	8	9	10	