

CI-0116 Análisis de Algoritmos y Estructuras de Datos
II ciclo de 2018

I EXAMEN PARCIAL

Jueves 20 de setiembre

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 debe indicar en el cuadro 1 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.

Cuadro 1: Información sobre las preguntas.

Pregunta	Puntos	Páginas	Calificación
1. <i>Algoritmos recursivos</i>	25		
2. <i>Algoritmos iterativos</i>	40		
3. <i>Solución de recurrencias</i>	15		
4. <i>Ordenamiento por selección</i>	8		
5. <i>Ordenamiento por inserción</i>	8		
6. <i>Ordenamiento por mezcla</i>	8		
7. <i>Ordenamiento por montículos</i>	12		
8. <i>Ordenamiento rápido</i>	8		
9. <i>Ordenamiento por residuos</i>	24		
Total	148		

Cuadro 2: Impuesto al valor agregado propuesto para distintos bienes y servicios, según el expediente N.º 20 580 la Asamblea Legislativa del 11 de setiembre de 2018. Se muestran en orden de aparición en el expediente.

Bien o Servicio	Impuesto	Artículo
General	13 %	10
Vuelos nacionales	4 %	11.1.a
Vuelos internaciones	10 %	11.1.a
Salud	4 %	11.1.b
Medicamentos	2 %	11.2.a
Educación ^a	0 %	11.2.b
Seguros	2 %	11.2.c
Agro	1 %	11.3.a
Canasta Básica	1 %	11.3.b

^aInicialmente se consignó un impuesto del 2% a la educación (privada), pero en mociones posteriores a la publicación del borrador se eliminó este impuesto.

Parte I

El *impuesto al valor agregado* (IVA) es una modalidad de recolección de impuestos usada en varias partes del mundo y que se está tratando de introducir en nuestro país. A diferencia del impuesto sobre las ventas, en el que el cobro del impuesto lo hace el vendedor final (el que vende el bien o servicio al consumidor), con el IVA el cobro se hace a lo largo de la cadena de producción y distribución, como se ejemplifica en la figura 1 para un caso ficticio de una mesa.

1. Algoritmos recursivos. [25 pts.]

- a) Escriba un método en C o C++ que, dada una tasa de IVA (por ejemplo, 0.15 en el ejemplo de la figura 1) y un arreglo de n precios cobrados en la cadena de elaboración y venta de un producto (por ejemplo, ₡1150, ₡3450 y ₡4600 en el mismo ejemplo), muestre en la consola el monto que debe pagar al fisco cada uno de los eslabones de la cadena. Por ejemplo, para la cadena de la figura 1 se puede mostrar algo así (en el orden cronológico):

Eslabón 0: 150 colones

Eslabón 1: 300 colones

Eslabón 2: 150 colones

o así (en el orden cronológico inverso) :

Eslabón 2: 150 colones

Eslabón 1: 300 colones

Eslabón 0: 150 colones

o en cualquier otro orden, siempre y cuando se especifique quién paga cuánto. Use el siguiente encabezado para el método:

```
void IVArec(float iva, float precios[], int n);
```

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

- b) Escriba una fórmula recursiva para el tiempo de ejecución 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 la parte a) es correcta]

2. Algoritmos iterativos. [40 pts.]

- a) Escriba una versión iterativa, `IVAit`, del algoritmo de la pregunta 1. [15 pts.]
- 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. [Válido solo si la parte a) es correcta].
- c) Escriba una fórmula para el tiempo de ejecución del algoritmo [5 pts.] y determine una cota asintótica lo más ajustada posible [5 pts.]. [Válido solo si la parte a) es correcta].

1. Quien produce la mesa

Compra insumos para hacer la mesa (madera, clavos, barniz).

Elabora la mesa aportando el valor de su trabajo y conocimiento y la vende a una tienda al detalle.



2. Tienda vende la mesa

Compra la mesa al productor y fija un precio de venta de €1000 adicionales.



3. Quien compra la mesa

Compra la mesa al precio de venta que ya incluye el IVA.



$$\begin{array}{ccccccc} \text{€1.000} + \text{€150} & = & \text{€1.150} & + & \text{€2.000} + \text{€300} & = & \text{€3.450} & + & \text{€1.000} + \text{€150} & = & \text{€4.600} & = & \text{€4.600} \\ \text{IVA} & & & & \text{IVA} & & & & \text{IVA} & & & & \end{array}$$

Figura 1: Ejemplo del proceso de cobro del *impuesto al valor agregado* (IVA) en una cadena de producción y venta de una mesa con un impuesto de 15%. El fabricante de la mesa compra los insumos por ₡1000 y paga al vendedor de estos un IVA de ₡150 (15% de ₡1000). Su trabajo de carpintería *agrega* un valor de ₡2000 al producto y al venderlo a una mueblería cobra a esta un IVA de ₡300 (15% de ₡2000). Por último, la mueblería añade al precio ₡1000 (lo que ella considera necesario por haber almacenado y hecho llegar el producto al consumidor) y cobra al consumidor un IVA de ₡150 (15% de ₡1000). Dentro de un plazo determinado por el fisco, el facilitador de los insumos, el carpintero y el vendedor de muebles entregan al fisco el IVA recolectado: ₡150, ₡300 y ₡150, respectivamente. (Extracto de infograma tomado del Semanario Universidad: <https://semanariouniversidad.com/pais/el-iva-ira>).

3. *Solución de recurrencias.* [15 pts.]

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

a) $T(n) = 4T(n/\sqrt{2}) + cn^3$. [5 pts.]

b) $T(n) = T(\sqrt{n}) + c$. [10 pts.]

Parte II

En las preguntas a continuación se pide ordenar *ascendentemente* los bienes y servicios del cuadro 2 *según el impuesto al valor agregado asignado a ellos. Si dos bienes o servicios tienen el mismo impuesto, deben ordenarse alfabéticamente.*

4. *Ordenamiento por selección.* [8 pts.]

Simule la ejecución del algoritmo de ordenamiento por selección sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada iteración del ciclo principal (externo). Si no muestra el estado de una casilla se asume que conserva el valor de la iteración anterior. Después del primer estado incorrecto, el resto no suman puntos.

	Posición								
Itn.	1	2	3	4	5	6	7	8	9
	Gral. (13%)	V. n. (4%)	V. i. (10%)	Salud (4%)	Meds. (2%)	Educ. (0%)	Segs. (2%)	Agro (1%)	C. b. (1%)
1									
2									
3									
4									
5									
6									
7									
8									

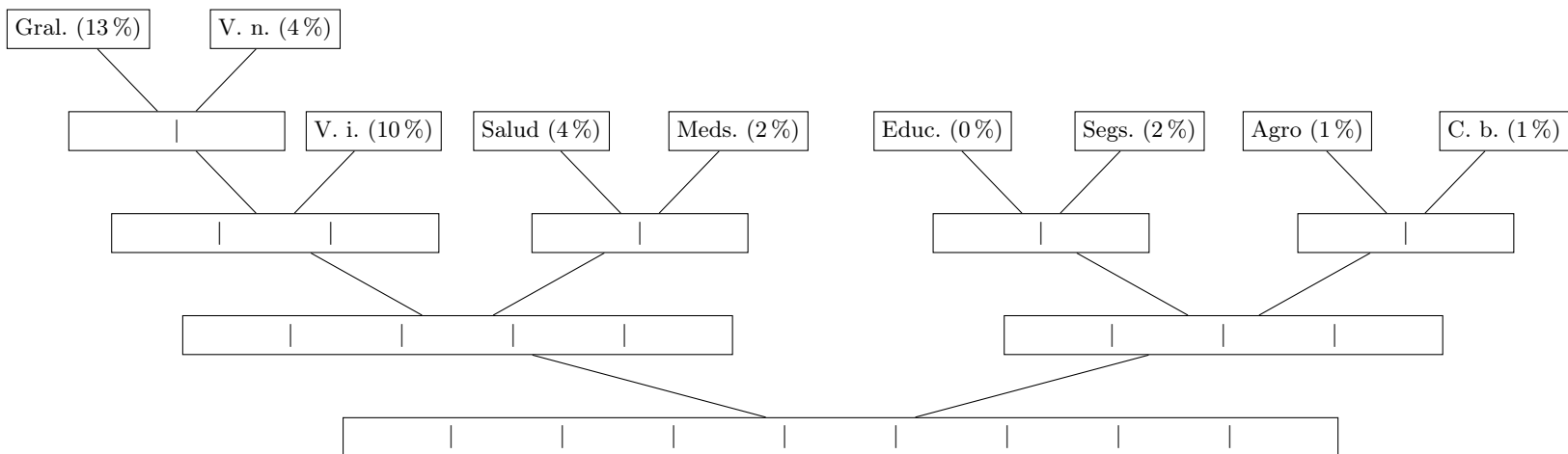
5. *Ordenamiento por inserción.* [8 pts.]

Simule la ejecución del algoritmo de ordenamiento por inserción sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada iteración del ciclo principal (externo). Si no muestra el estado de una casilla se asume que conserva el valor de la iteración anterior. Después del primer estado incorrecto, el resto no suman puntos.

	Posición								
Itm.	1	2	3	4	5	6	7	8	9
	Gral. (13%)	V. n. (4%)	V. i. (10%)	Salud (4%)	Meds. (2%)	Educ. (0%)	Segs. (2%)	Agro (1%)	C. b. (1%)
2									
3									
4									
5									
6									
7									
8									
9									

6. *Ordenamiento por mezcla.* [8 pts.]

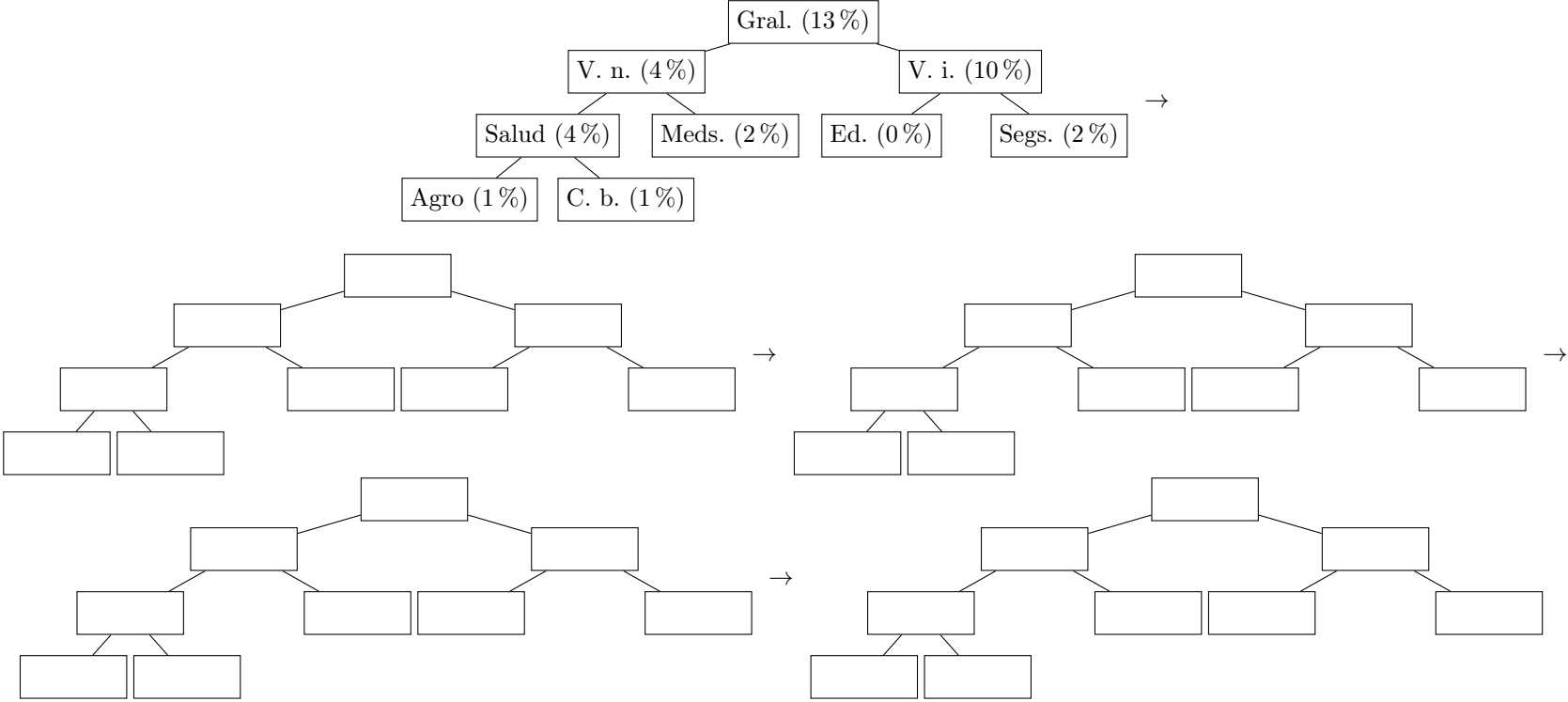
Simule la ejecución del algoritmo de ordenamiento por mezcla sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada mezcla. Después de la primer mezcla incorrecta el resto de mezclas no suman puntos.



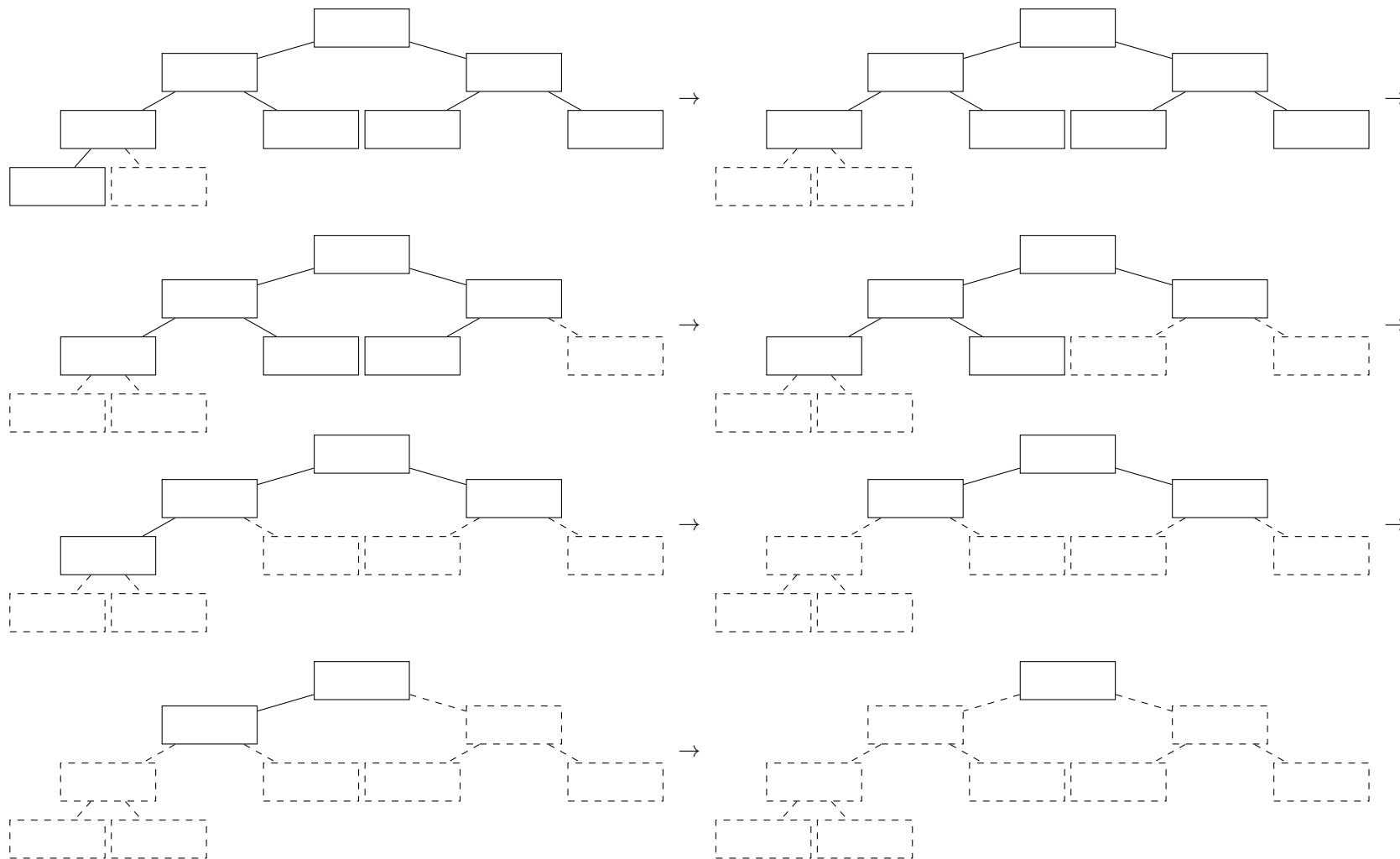
7. *Ordenamiento por montículos.* [12 pts.]

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 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, pues su contenido es trivial). Después de la primer operación incorrecta el resto de operaciones no suman puntos.

MONTICULARIZAR:



ORDENAR:



8. *Ordenamiento rápido*. [8 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 de la iteración anterior. Después del primer estado incorrecto el resto no suman puntos.

Llamado	Estado del arreglo									Pos. piv.
	1	2	3	4	5	6	7	8	9	
	Gral. (13%)	V. n. (4%)	V. i. (10%)	Salud (4%)	Meds. (2%)	Educ. (0%)	Segs. (2%)	Agro (1%)	C. b. (1%)	
1.º	1	2	3	4	5	6	7	8	9	
2.º	1	2	3	4	5	6	7	8	9	
3.º	1	2	3	4	5	6	7	8	9	
4.º	1	2	3	4	5	6	7	8	9	
5.º	1	2	3	4	5	6	7	8	9	
6.º	1	2	3	4	5	6	7	8	9	
7.º	1	2	3	4	5	6	7	8	9	
8.º	1	2	3	4	5	6	7	8	9	
9.º	1	2	3	4	5	6	7	8	9	

9. *Ordenamiento por residuos.* [24 pts.]

Simule la ejecución del algoritmo de ordenamiento por residuos para los datos del cuadro 2 *tomando el impuesto como la parte más significativa y el nombre como la menos significativa*. Muestre en el cuadro 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 [9 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.

A		
1	Gral.	13%
2	V. n.	4%
3	V. i.	10%
4	Salud	4%
5	Meds.	2%
6	Educ.	0%
7	Segs.	2%
8	Agro	1%
9	C. b.	1%

C :

C' :

C'' :

$B \rightarrow A$	
1	
2	
3	
4	
5	
6	
7	
8	
9	

C :

C' :

C'' :

B	
1	
2	
3	
4	
5	
6	
7	
8	
9	