

**CI-1221 Estructuras de Datos y Análisis de Algoritmos**  
**II ciclo de 2017**

**I EXAMEN PARCIAL**

*Jueves 14 de setiembre, 12:15 p. m. – 2.45 p. m.*

Nombre: \_\_\_\_\_ Carné: \_\_\_\_\_

El examen consta de 11 preguntas que suman 183 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>	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>Búsqueda de la mediana</i>	9		
10. <i>Ordenamiento por residuos</i>	20½		
11. <i>Árboles de decisión</i>	18½		
Total	183		

Cuadro 2: Diez huracanes que han producido más pérdidas económicas para los EE. UU. (Los costos de los más recientes —Irma y Harvey— se basan en estimados preliminares).

Huracán	Año	Estado	Costo (millardos de dólares)
Katrina	2005	Luisiana (LA)	160
Irma	2017	Florida (FL)	110
Harvey	2017	Texas (TX)	90
Sandy	2012	Nueva Jersey (NJ)	70
Andrew	1992	Florida (FL)	48
Ike	2008	Texas (TX)	35
Iván	2004	Alabama (AL)	27
Wilma	2005	Florida (FL)	24
Rita	2005	Texas (TX)	24
Charley	2004	Florida (FL)	21

## Parte I

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

Sea  $B$  una matriz de booleanos (valores *cero* o *uno*) de tamaño  $m \times n$ . Se desea recorrerla desde la esquina superior izquierda a la inferior derecha visitando la mayor cantidad posible de *unos*, sujeto a que en cada paso solo se puede mover a la casilla derecha o a la inferior.

Lupe la Miope ha divisado un algoritmo recursivo para, según ella, visitar la mayor cantidad de *unos*. Su algoritmo es el siguiente:

- I) Si hay un *uno* a la derecha o abajo, visítelo; sino, decida arbitrariamente dónde moverse (a la derecha o abajo).
- II) Repita el paso I hasta llegar a la esquina inferior derecha.

- a) Implemente una función recursiva en C o C++ que devuelva la cantidad de *unos* visitados siguiendo la estrategia de Lupe. Use el siguiente encabezado:

```
int LupeRecursiva(int B[][], int m, 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.]
- c) Determine una cota asintótica para el tiempo de ejecución del algoritmo. [5 pts.]

### 2. Algoritmos iterativos. [40 pts.]

- a) Escriba una versión iterativa del algoritmo de Lupe implementado en la pregunta 1. [15 pts.]
- b) Establezca la correctitud del algoritmo mostrando que cumple la estrategia de Lupe. 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 [5 pts.] y determine una cota asintótica lo más ajustada posible [5 pts.].

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

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

- a)  $T(n) = c^2 T(n/c) + c \lg(n^2)$ . [10 pts.]
- b)  $T(n) = T(n-1) + 2^{n-1}$ . [10 pts.]

## Parte II

Los problemas de esta parte tratan sobre los huracanes que más costo económico han causado a los EE. UU y que se muestran en el cuadro 2.<sup>1</sup> En ellos se pide reordenar los huracanes en orden alfabético usando varios algoritmos. Puede usar abreviaturas siempre que no afecten el ordenamiento.

---

<sup>1</sup>Datos tomados de [www.businessinsider.com/hurricane-irma-costliest-hurricanes-us-history-map-2017-9](http://www.businessinsider.com/hurricane-irma-costliest-hurricanes-us-history-map-2017-9) y en [en.wikipedia.org/wiki/List\\_of\\_costliest\\_Atlantic\\_hurricanes](https://en.wikipedia.org/wiki/List_of_costliest_Atlantic_hurricanes).

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

Simule la ejecución del algoritmo de ordenamiento por selección 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 de inserciones no suman puntos.

	Posición									
Itn.	1	2	3	4	5	6	7	8	9	10
	Katrina	Irma	Harvey	Sandy	Andrew	Ike	Iván	Wilma	Rita	Charley
1										
2										
3										
4										
5										
6										
7										
8										
9										

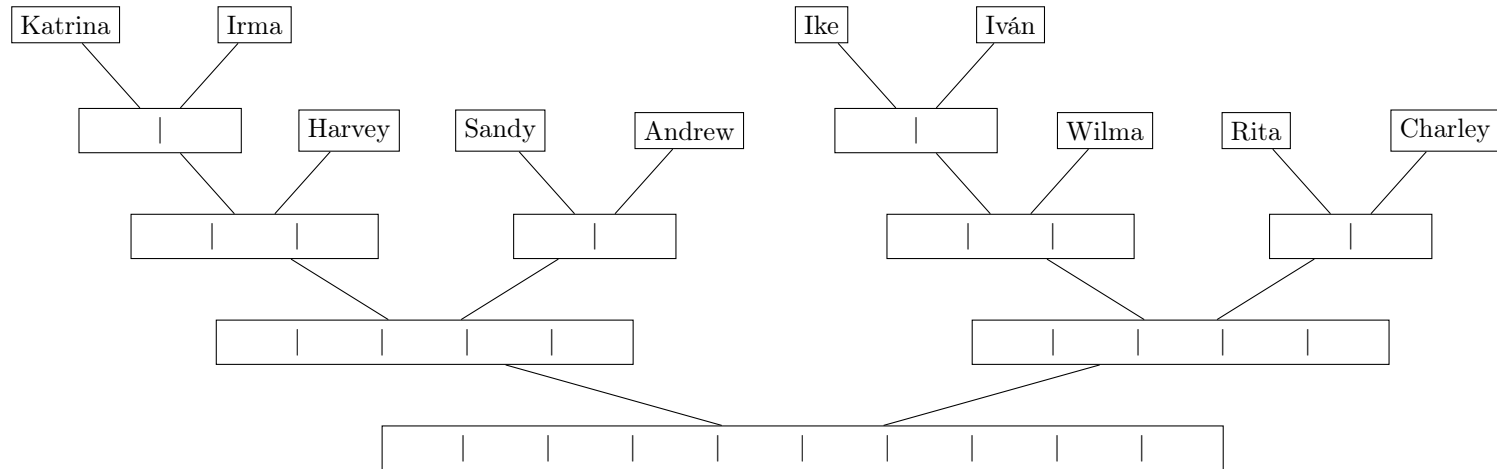
5. *Ordenamiento por inserción.* [9 pts.]

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 de inserciones no suman puntos.

	Posición									
Itn.	1	2	3	4	5	6	7	8	9	10
	Katrina	Irma	Harvey	Sandy	Andrew	Ike	Iván	Wilma	Rita	Charley
2										
3										
4										
5										
6										
7										
8										
9										
10										

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

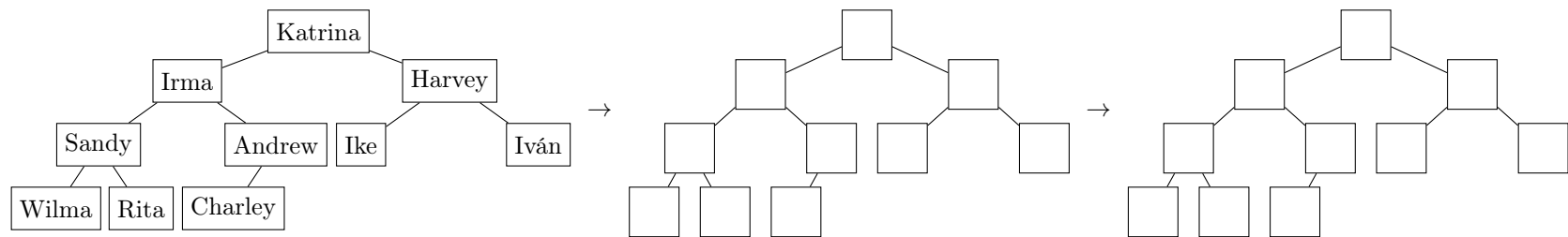
Simule la ejecución del algoritmo de ordenamiento por mezcla sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada una de las mezclas. Si no muestra el estado de una casilla se asume que conserva el valor que tenía en la iteración anterior. Después de la primer mezcla incorrecta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

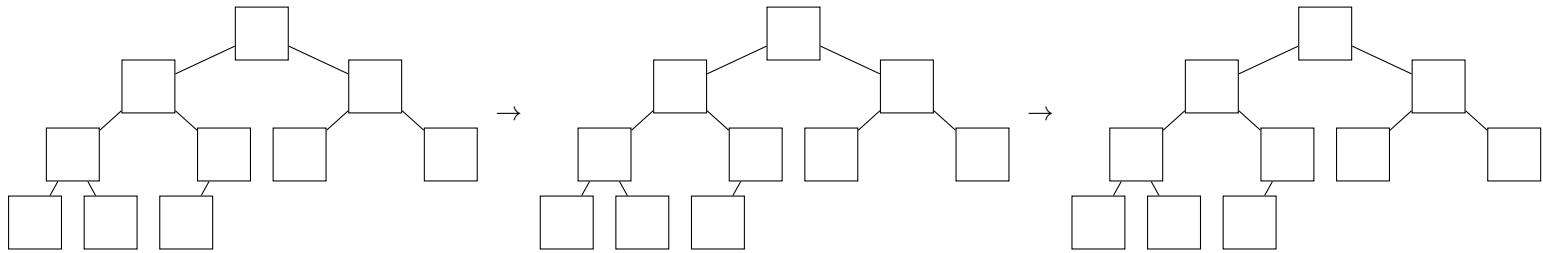


7. *Ordenamiento por montículos.* [14 pts.]

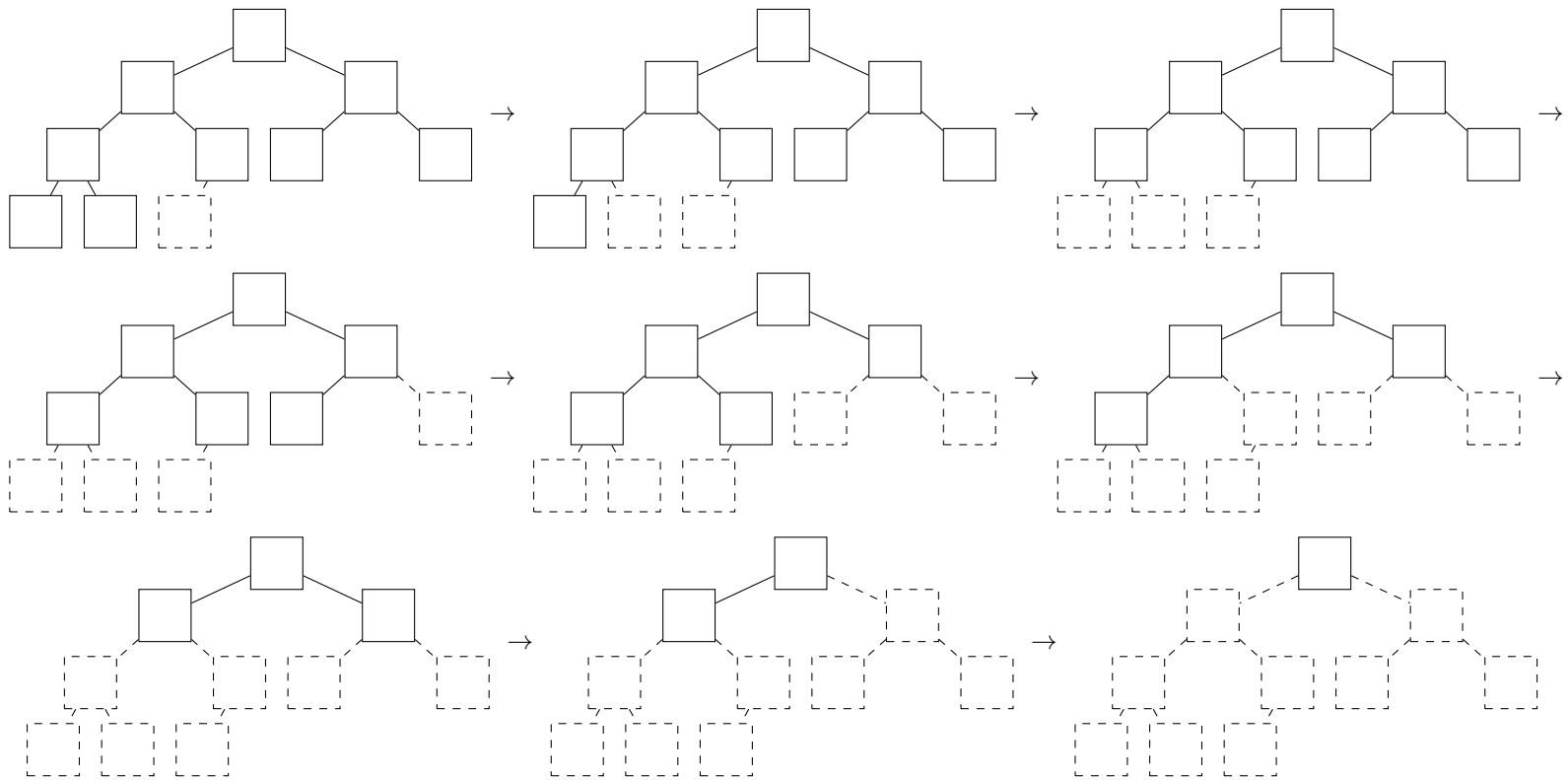
Simule la ejecución del algoritmo de ordenamiento por montículos sobre el arreglo mostrado abajo en forma de árbol. Muestre el estado del *montículo* después de cada llamado a CORREGIR-CIMA y EXTRAER-MÁXIMO. Si deja un nodo vacío, se asume que tiene el mismo valor que en el paso anterior. (Los nodos 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.

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 de estados no suman puntos.

Llamado	Estado del arreglo										Posición del pivote
	1	2	3	4	5	6	7	8	9	10	
	Katrina	Irma	Harvey	Sandy	Andrew	Ike	Iván	Wilma	Rita	Charley	
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. *Búsqueda de la mediana.* [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	
	Katrina	Irma	Harvey	Sandy	Andrew	Ike	Iván	Wilma	Rita	Charley	
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	

10. *Ordenamiento por residuos.* [20½ pts.]  
 Simule la ejecución del algoritmo de ordenamiento por residuos para los códigos de los estados donde alcanzaron tierra los huracanes, conservando como información *satélite* el nombre del huracán, o su inicial, mostrado como subíndice: Katrina: LA<sub>K</sub>; Irma: FL<sub>I</sub>; Harvey: TX<sub>H</sub>; Sandy: NJ<sub>S</sub>; Andrew: FL<sub>A</sub>; Ike: TX<sub>I</sub>; Iván: AL<sub>I</sub>, Wilma: FL<sub>W</sub>; Rita: TX<sub>R</sub>; y Charley: FL<sub>C</sub>. Muestre lo siguiente para cada llamado a la subrutina de ordenamiento por conteo: el histograma *C* [3 pts.], el histograma acumulativo *C'* [3 pts.], el arreglo resultante *B* [10 pts.] y el estado final del histograma acumulativo *C''* (después de producir el arreglo *B*) [4½ pts.]. Después del primer arreglo incorrecto el resto de arreglos no suman puntos.

$A$		$C:$				$B \rightarrow A$		$C:$					$B$	
1	LA <sub>K</sub>	A J L X				1		A F L N T					1	
2	FL <sub>I</sub>	A J L X				2		A F L N T					2	
3	TX <sub>H</sub>					3							3	
4	NJ <sub>S</sub>					4							4	
5	FL <sub>A</sub>					5							5	
6	TX <sub>I</sub>					6							6	
7	AL <sub>I</sub>					7							7	
8	FL <sub>W</sub>					8							8	
9	TX <sub>R</sub>					9							9	
10	FL <sub>C</sub>					10							10	

11. *Árboles de decisión.* [18½ pts.]

Complete el siguiente árbol de decisión con las comparaciones hechas por el algoritmo de ordenamiento *descendiente* por selección —es decir, en la  $i$ -ésima iteración el algoritmo selecciona el mayor de los que quedan y lo pone en la posición  $i$ —, para el arreglo  $A = \langle a_1, a_2, a_3 \rangle$  [1½ pts. cada nodo interno, 1 pto. cada hoja]. Si una comparación es incorrecta, tanto ella como su descendencia no suman puntos.

