CI-1221 Estructuras de Datos y Análisis de Algoritmos
II ciclo 2015, grupos 3 y 4

I Examen Parcial

Sábado 26 de setiembre

Nombre: _____ Carné: _____

El examen consta de 9 preguntas que suman 165 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 es indicado en los subítemes. Muchas de ellas tratan sobre las atracciones «juveniles» del Parque Nacional de Diversiones, mostradas abajo. 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 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. *Algoritmos recursivos*	25		
2. *Algoritmos iterativos*	45		
3. *Solución de recurrencias*	15		
4. *Ordenamiento por selección*	9		
5. *Ordenamiento por inserción*	9		
6. *Ordenamiento por mezcla*	9		
7. *Ordenamiento por montículos*	14		
8. *Ordenamiento rápido*	9		
9. *Ordenamiento por residuos*	30		
Total	165		

1. *Algoritmos recursivos.* [25 pts.]
   Clemente *el Impaciente* ha divisado un algoritmo recursivo para, según el, visitar las n atracciones «juveniles» del parque de Diversiones de la forma más rápida posible. Su algoritmo es el siguiente:

   Si queda por visitar solo una atracción:

   - Diríjase a esa atracción.

   Si quedan por visitar $n > 1$ atracciones:

   - Diríjase a la atracción más cercana aún no visitada y luego escoja la siguiente usando este mismo algoritmo.

   a) Implemente un algoritmo recursivo en C o C++ que despliegue el orden en que, según Clemente, se deben visitar las n atracciones juveniles del parque para hacerlo de la forma más rápida. Para ello se *recomienda* usar el siguiente encabezado, pero puede usar otro si lo prefiere:

   ```
   void ClementeRecursivo(int n, int distancia[][], bool visitado[], int nVisitados, int actual=0);
   ```

   donde `distancia[0][j]` es la distancia desde la entrada/salida del parque a la atracción j ($j = 1, \ldots, n$), `distancia[i][j]` es la distancia entre las atracciones i y j ($i = 1, 2, \ldots, n$ y $j = 1, 2, \ldots, n$), `visitado[i]` indica si la atracción i ya ha sido visitada o no (puede asumir que fue inicializada con falsos), `nVisitados` indica la cantidad de atracciones que han sido visitadas (cero inicialmente) y `actual` indica la atracción en la que se encuentra en este momento (`actual=0`, inicialmente). Asuma que `distancia[i][j]` es igual a `distancia[j][i]` para $i = 0, 1, \ldots, n$ y $j = 0, 1, \ldots, n$. Puede usar métodos auxiliares si lo necesita. [15 pts.]

   b) Escriba una fórmula recursiva para el tiempo de ejecución del algoritmo. [3 pts.]

   c) Determine una cota asintótica para el tiempo de ejecución del algoritmo. [7 pts.]

2. *Algoritmos iterativos.* [45 pts.]

   a) Escriba un algoritmo iterativo que produzca la misma secuencia de atracciones que el algoritmo «ClementeRecursivo» mostrado en la pregunta 1. [15 pts.]

   b) Establezca la correctitud del algoritmo mostrando que cumple el criterio de Clemente. Para ello identifique el invariante del ciclo (más externo) [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.* [15 pts.]
   Resuelva las siguientes recurrencias asumiendo que $T(n) = \Theta(1)$ para $n \leq 1$ y que $k > 1$.

   a) $T(n) = kT(n/k) + k \lg(n/k)$. [7 pts.]

   b) $T(n) = 3\,T(2n/3) + 3n^2$. [8 pts.]

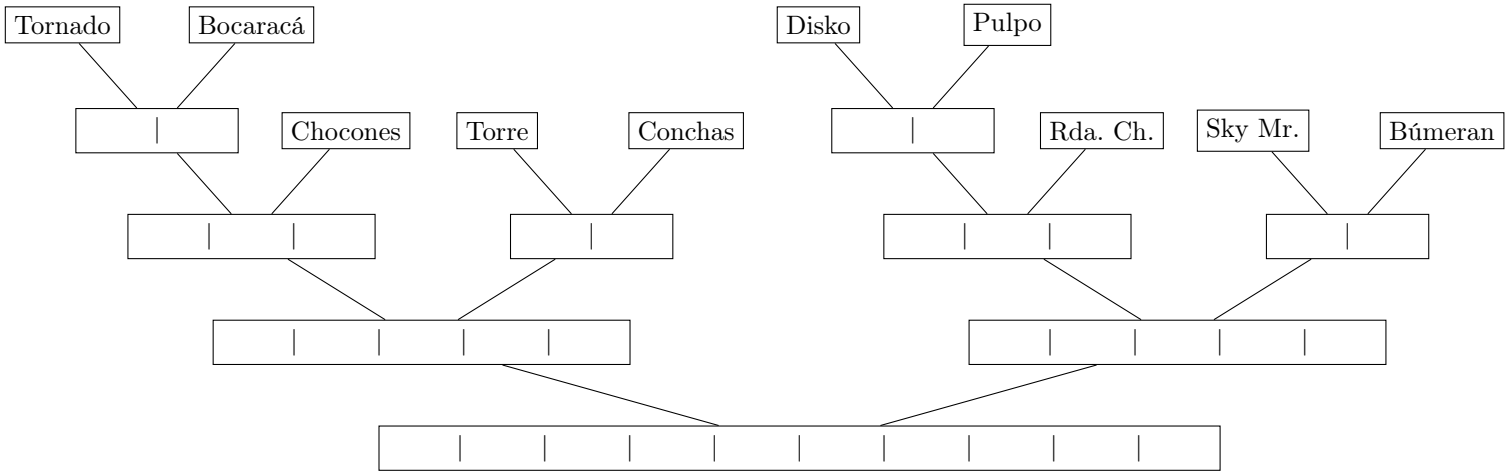
4. *Ordenamiento por selección.* [9 pts.]
   Simule la ejecución del algoritmo de ordenamiento por selección sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada una de las iteraciones del ciclo principal (externo). Si no muestra el estado de una casilla se asume que conserva el mismo valor que 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
0	Tornado	Bocaracá	Chocones	Torre	Conchas	Disko	Pulpo	Rda. Ch.	Sky Mr.	Búmeran
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 mismo valor que en la iteración anterior. Después de la primer inserción incorrecta el resto de inserciones no suman puntos.

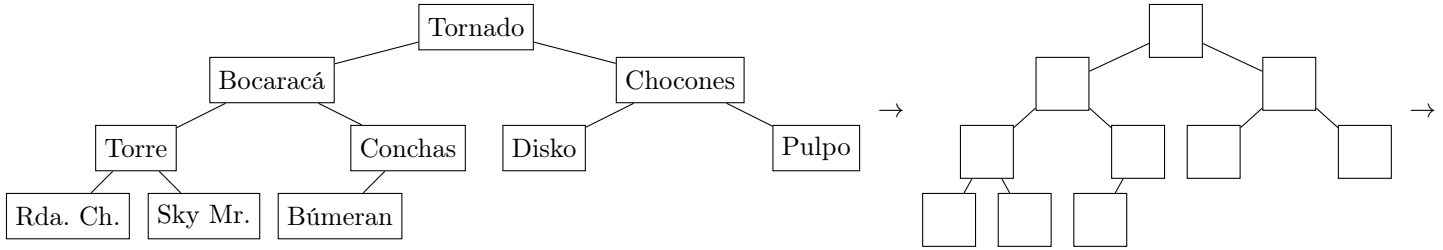
	Posición									
Itn.	1	2	3	4	5	6	7	8	9	10
1	Tornado	Bocaracá	Chocones	Torre	Conchas	Disko	Pulpo	Rda. Ch.	Sky Mr.	Búmeran
2										
3										
4										
5										
6										
7										
8										
9										
10										

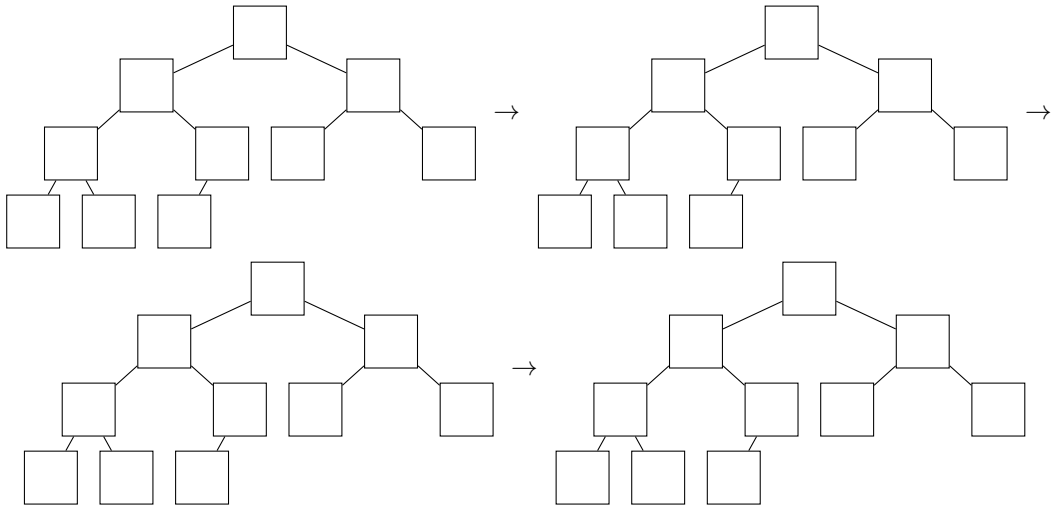
6. *Ordenamiento por mezcla.* [9 pts.]
   Simule la ejecución del algoritmo de ordenamiento por mezcla sobre el arreglo mostrado abajo. Muestre el estado del arreglo al finalizar cada una de las mezclas. Si no muestra el estado de una casilla se asume que conserva el mismo valor que en la iteración anterior. Después de la primer mezcla incorrecta el resto de mezclas no suman puntos.



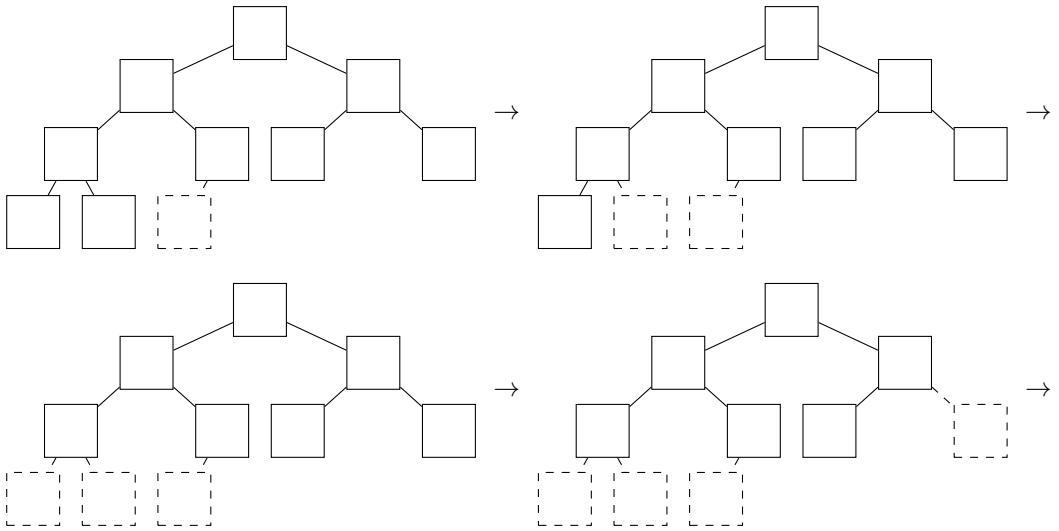
7. *Ordenamiento por montículos.* [14 pts.]
   Simule la ejecución del algoritmo de ordenamiento por montículos sobre el arreglo mostrado abajo en forma de árbol. Muestre el estado del *montículo* después de cada llamado a Corregir-Cima y Extraer-Máximo. Si deja un nodo vacío, se asume que tiene el mismo 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.

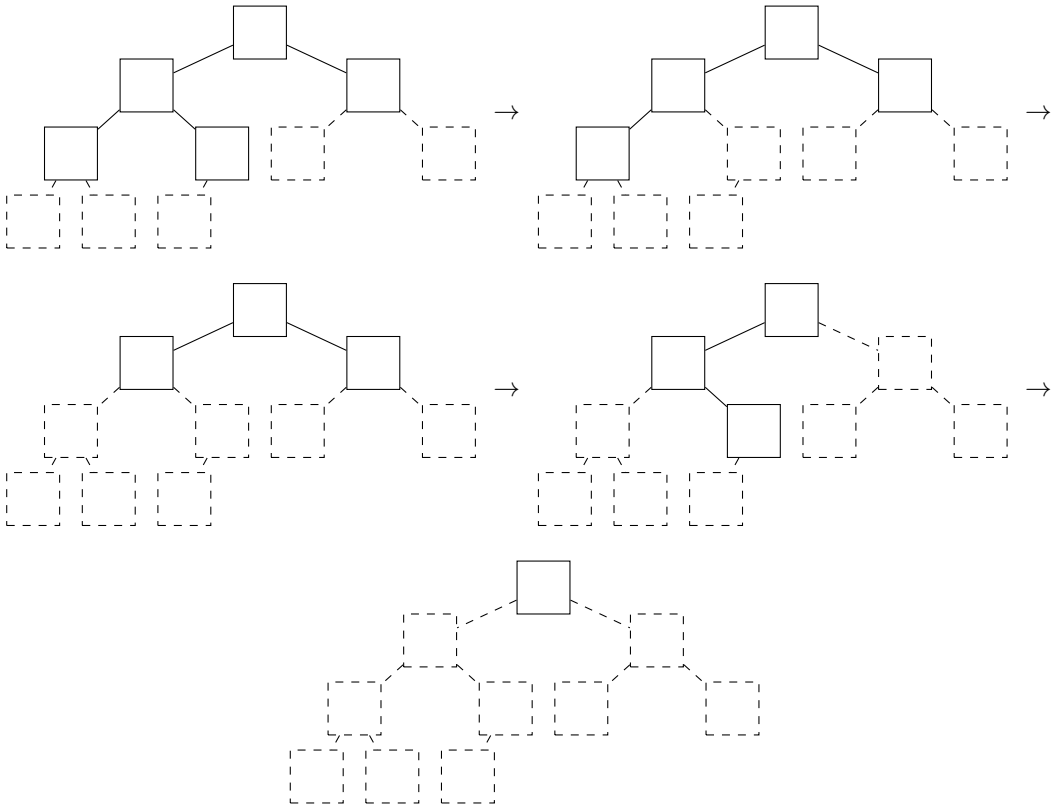
   Monticularizar:





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 mismo valor que 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	
	Tornado	Bocaracá	Chocones	Torre	Conchas	Disko	Pulpo	Rda. Ch.	Sky Mr.	Búmeran	
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.* [30 pts.]

Simule la ejecución del algoritmo de ordenamiento por residuos sobre las siguientes abreviaturas de las atracciones juveniles del Parque de Diversiones: TND (Tornado), BCC (Bocaracá), CHC (Chocones), TRR (Torre), CCH (Conchas), DSK (Disko), PLP (Pulpo), RCH (Rueda de Chicago), SMR (Sky Master) y BMR (Búmeran). Muestre lo siguiente para cada llamado a la subrutina de ordenamiento por conteo:

I. El histograma C. [5 pts.]

II. El histograma acumulativo C'. [5 pts.]

III. El arreglo resultante B. [15 pts.]

IV. El estado final del histograma acumulativo C'' (después de producir el arreglo B). [5 pts.]

Después del primer arreglo incorrecto el resto de arreglos no suman puntos.