

UNIVERSIDAD DE COSTA RICA FACULTAD DE INGENIERÍA ESCUELA DE CIENCIAS DE LA COMPUTACIÓN E INFORMÁTICA

CI-1221 Estructuras de Datos y Análisis de Algoritmos I Ciclo de 2010

II EXAMEN PARCIAL

Martes 4 de abril, 5:00 p.m. a 7:30 p.m.

El examen consta de 125 puntos pero no se reconocerán más de 120. Cada pregunta empieza con un indicador de su puntaje y del tema del que trata. Si la pregunta tiene subítemes, el puntaje de cada uno de ellos es indicado al final de la pregunta. Se recomienda echar un vistazo a los temas de las preguntas y a su puntaje antes de empezar a resolver el examen para así distribuir su tiempo y esfuerzo de la mejor manera, de acuerdo a sus destrezas.

1. [10 pts.] *Listas doblemente enlazadas con centinela*. Escriba el (pseudo) código para insertar un nodo *x* en una lista doblemente enlazada *L* con nodo centinela *L.nil* (o *nil*[*L*]). Utilice para ello el siguiente encabezado:

ListInsert(L,x)

- 2. [25 pts.] *Tablas hash*. Sea una tabla hash con resolución de colisiones mediante encadenamiento.
 - (a) Utilice la tabla siguiente para indicar la complejidad temporal de inserciones, borrados y búsquedas fallidas y exitosas, en el peor y mejor casos, y en el caso promedio, asumiendo que la función hash cumple con la propiedad de *distribución uniforme de los elementos*. [1 pto. c/celda vacía].

Caso →	Peor	Promedio	Mejor
Operación ↓			
Inserción			
Borrado			
Búsqueda fallida			
Búsqueda exitosa			$\Theta(1)$

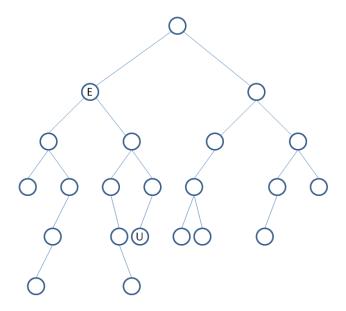
(b) Repita el paso a pero esta vez asumiendo que cada cadena (lista enlazada) se mantiene ordenada [1 pto. c/celda vacía].

	Caso →	Peor	Promedio	Mejor
Búsqueda	\downarrow			
Inserción				
Borrado				
Fallida				
Exitosa				$\Theta(1)$

- (c) Compare las tablas y determine si vale la pena mantener las cadenas ordenadas. Explique. [3 pts.]
- 3. [10 pts.] *Tablas hash*. Sea una tabla hash de tamaño 10 con resolución de colisiones mediante direccionamiento abierto (*open addressing*) y sondeo cuadrático (*quadratic probing*) $h(k,i)=(k \mod 10+i^2)\mod 10$. Inserte los elementos 0, 10, 20, 30 y 40 en la tabla y muestre la configuración de la tabla después de cada una de las inserciones utilizando la tabla siguiente (si lo prefiere puede mostrar en cada fila solo el elemento recién insertado). [2 pts. c/ inserción]

$\frac{\text{Posición}}{\text{Elemento}} \rightarrow$ $\text{Insertado} \downarrow$	0	1	2	3	4	5	6	7	8	9
0										
10										
20										
30										
40										

4. [10 pts.] Árboles de búsqueda binarios. Determine el predecesor de E y el predecesor de U en el siguiente árbol de búsqueda binario. Se sugiere llamarlos D y T, respectivamente, pero puede llamarlos de otra manera siempre y cuando especifique claramente cuál es cuál.



- 5. [30 pts.] Árboles rojinegros.
 - (a) Inserte en un árbol rojinegro las llaves 1, 2, 3, 4, 5 y 6, en ese orden. Muestre en cada inserción todas las rotaciones (intercambios de rol de parternidad) y cambios de color que se efectúen. [la inserción del iésimo elemento vale i puntos (i = 1,2,...,6).]
 - (b) Borre la llave 1 mostrando todos los pasos relevantes. [9 pts.]
- 6. [18 pts.] *Ampliación de árboles rojinegros*. Se desean ampliar los nodos de un arbol rojinegro con información acerca de su altura negra.
 - (a) Un requisito para que la altura negra de un nodo se pueda actualizar de forma eficiente después de una modificación al árbol es que se pueda determinar utilizando solamente la información contenida en el nodo y en sus hijos. ¿Será esto posible? Explique. [8 pts.]
 - (b) Escriba el segmento del (pseudo) código que permite actualizar la altura negra de un nodo. [10 pts.]
- 7. [20 pts.] Ampliación de árboles de estadísticas de orden. Explique cómo se puede ampliar un árbol de estadísticas de orden para implementar las operaciones Minimum(L) [6 pts.] y Successor(L,x) [6 pts.] en tiempo Θ(1) sin que se vean afectados los órdenes de duración de las otras operaciones. Argumente el por qué de la perseverancia de los órdenes de duración de las otras operaciones [8 pts].