

Tarea 6

Profesores: Armando Castaeda y Sergio Rajsbaum

Ayudantes: Luis Gmez y Diego Velzquez

Martes 22 de octubre de 2019

Fecha de entrega: Jueves 29 de octubre de 2019

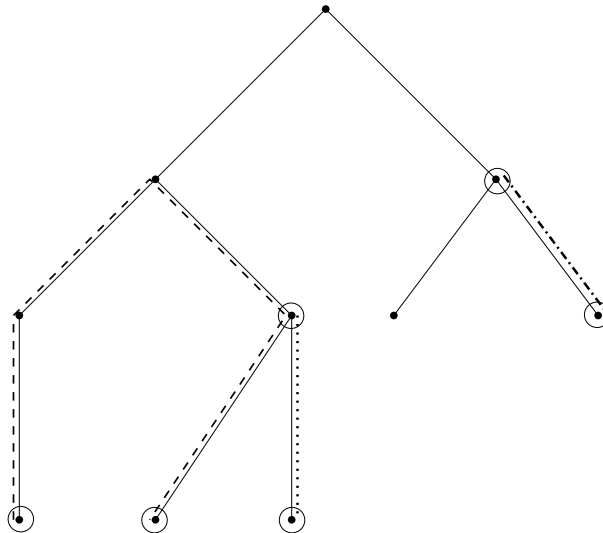
INSTRUCCIONES:

- Se puede hacer en equipos de 2 o 3 personas, pero hay que entregarla individualmente. La resuelven entre todos, pero cada quien la escribe en sus palabras. Anotar en cada tarea el nombre de todos los miembros del equipo.
- Las respuestas deben estar escritas con claridad, todos los enunciados demostrados.
- No se aceptan tareas después de la fecha límite.
- No escribas la implementación de tus algoritmos.
- Si el ejercicio dice “prueba”, “demuestra” o “muestra”, no debes dejar ningún hecho sin justificar; esto significa que debes decir por qué lo que escribes es verdad. Si lo que se pide es una explicación, es suficiente que enuncies los hechos que explican lo que se pide sin decir por qué son verdaderos pero tienes que manifestarlos completamente.

PROBLEMAS:

1. Sea T un árbol binario con raíz y S un conjunto con un número par de vértices de T . Diseña un algoritmo de tipo *divide-y-vencers* de tiempo $O(n)$ que empareje los nodos en S de forma tal que en T existan caminos disjuntos en aristas entre cada par.

La figura siguiente muestra un ejemplo del problema en el que S tiene los seis vértices encerrados en círculos y los emparejamientos se denotan con tres tipos distintos de líneas punteadas.



2. Considera las siguientes recurrencias para las que $T(1) = c$:
 - a) $T(n) = T(n/2) + O(1)$. Prueba que $T(n) = \Theta(\log n)$.
 - b) $T(n) = T(n/2) + O(n)$. Prueba que $T(n) = O(n)$.
 - c) $T(n) = T(n/d) + T(n(d-1)/d) + O(n)$, donde d es un entero positivo. Prueba que $T(n) = O(n \log n)$.

3. Se tienen dos bases de datos, BD_1 y BD_2 , cada una con n valores numéricos distintos, en total $2n$ valores distintos. La única manera de acceder los datos es mediante preguntas, en las cuales se le da un valor $i \in \{1, \dots, n\}$ a una de las dos bases de datos, BD_j , y esta responde con el i -ésimo valor más pequeño que contiene BD_j . Diseña un algoritmo que dado un entero $k \in \{1, \dots, 2n\}$ de entrada, encuentre el k -ésimo valor más pequeño en las bases de datos, usando $O(\log n)$ preguntas. Demuestra la correctez y complejidad de tu algoritmo.

4. En clase revisamos el algoritmo de los códigos de Huffman. En el primer intento de prueba, quisieron probar que el árbol T que generaba el código encontrado satisfacía:
 - a) T es completo.
 - b) Si a y b son hojas del árbol tales que $\text{profu}(a) < \text{profu}(b)$ entonces $f_a \geq f_b$ (f_x es la frecuencia con que aparece x en el texto).
 - c) Los dos símbolos de frecuencia menor son hermanos en T .

Demuestra que esa estrategia no funciona para probar lo que queríamos. Es decir, ofrece un alfabeto S con frecuencias $f : S \rightarrow [0, 1]$ y encuentra un código φ que no sea óptimo y cuyo árbol generado satisfaga las tres propiedades anteriores. Demuestra por qué el código que propones no es óptimo.

Nota: La prueba vista en clase de hecho muestra que la propiedad arriba mencionada se cumple no solo para el árbol T con n hojas, sino para el árbol T' que junta a y b en un solo símbolo.