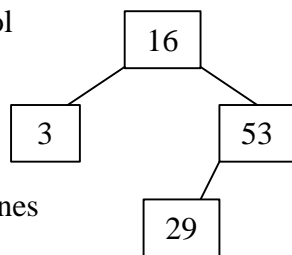


No entregar esta hoja con el examen. Te la puedes llevar.  
Todas las preguntas tienen la misma ponderación (25%).

2. Es bien conocido que las tablas de dispersión abiertas y las cerradas tienen distinta eficiencia en tiempo y uso de memoria. Suponer que queremos usar tablas de dispersión para almacenar mensajes cortos de texto. Cada mensaje va en un registro que ocupa siempre 256 bytes. Suponemos que los punteros ocupan 4 bytes. Considerar que el número de mensajes que vamos a almacenar es  $n$  y el número de cubetas de la tabla es  $B$ . Recordar que la eficiencia en dispersión cerrada es un  $O(1/(1-n/B))$ .
- Estimar cuánta memoria, en bytes, se necesita usando dispersión abierta y dispersión cerrada. Justificar la respuesta, incluyendo un dibujo de las tablas.
  - ¿En qué porcentaje de llenado (es decir,  $n/B$ ) ocupa más memoria la dispersión abierta o la cerrada?
  - Suponer que tanto en dispersión abierta como en cerrada queremos establecer el número de cubetas,  $B$ , de manera que el tamaño promedio de las secuencias de búsqueda sea 3. Si  $n$  vale 1000, ¿cuánto debe ser  $B$  en dispersión abierta y en dispersión cerrada?

3. Resolver las siguientes cuatro cuestiones.

a) En el siguiente árbol AVL, insertar los elementos 52 y 82. Mostrar los árboles resultantes y, si son necesarias, las rotaciones que se aplican.



c) En un árbol trie se insertan las palabras: uno, una, unos, dos, tres, trece. Mostrar el árbol trie resultante con la representación vista en clase (es decir, poniendo las etiquetas en las relaciones de un nodo con sus hijos).

b) En el árbol AVL resultante del apartado a) se eliminan los elementos 53 y 82. Mostrar los árboles resultantes y, si son necesarias, las rotaciones que se aplican.

d) En un árbol B de orden  $p=4$  se insertan los elementos: 38, 82, 23, 9, 45, 63, 1, 99, 8. Mostrar el árbol B resultante.

4. Tenemos un amigo escritor que es un poco patoso para escribir en el teclado. Cuando va a empezar a escribir y pulsa una tecla, luego para escribir la siguiente letra no levanta la mano, de manera que pulsa todas las teclas que haya por medio. Eso sí, siempre va por el camino más corto de una tecla a otra. Por ejemplo, para escribir "cm" podría escribir "cvbnm" o "cfghjm", pero aprovechando la tecla de espacio puede poner "c m".

Estamos diseñando un teclado especial para nuestro amigo. Suponer que tenemos en total  $n$  teclas, numeradas desde 1 hasta  $n$ ; para cada tecla, tenemos una lista de las teclas que tiene a su lado, es decir, las teclas que puede pulsar sin cometer errores (por ejemplo, si la tecla "a" tiene a su lado la "b", entonces puede escribir "ab" sin cometer errores). Queremos saber cuál será el máximo número de errores cometerá en nuestro teclado, en cualquier posible secuencia de dos de letras.

Escribir un algoritmo eficiente que dada la descripción del teclado (como se indica arriba), calcule el máximo error posible que puede cometer y el par de letras que produce ese error. Si hay varias combinaciones de letras posibles, puede mostrar cualquiera de ellas.