

## **TRABAJO PRACTICO N°4** **Tema: Arboles Tries**

---

1. Dibujar los Tries correspondientes para los siguientes conjuntos de palabras:
  - a) maestra, maestro, masa, mata, miel, miente, reloj, rojo, rueda, ruedo.
  - b) semana, casa, caso, canto, bota, canción, cantina, semanario, cantera, bote, cancan.
2. Implementar las operaciones básicas del TDA Trie usando:
  - a) Matriz.
  - b) Arreglos enlazados.
  - c) Listas enlazadas.Probar las implementaciones con distintos conjuntos de palabras.
3. Implementar las siguientes operaciones para un Trie:
  - a) Devolver la cantidad de palabras almacenadas en el Trie.
  - b) Listar todas las palabras almacenadas en el Trie.
  - c) Mostrar todas las palabras almacenadas en un Trie, que comienzan con un determinado prefijo.
  - d) Modificar el algoritmo buscar, para que devuelva palabras parecidas a la buscada, en caso que esta no exista en el Trie. Nota: dos palabras son parecidas si las primeras  $(n / 2)$  letras coinciden.
  - e) Contar la cantidad de prefijos distintos existentes en el Trie.
  - f) ¿Qué ocurre si en lugar de buscar por prefijos queremos buscar por sufijos?. Dar una idea de cómo resolver el problema en ese caso.
4. Generar un Trie a partir de un archivo de texto. Probar las distintas operaciones del ejercicio anterior sobre el Trie en memoria y al finalizar la ejecución, almacenar el contenido del Trie en un archivo de texto. Hacerlo palabra por palabra.
5. Implementar una *calculadora binaria* que realice las operaciones básicas de suma y resta. Guardar en un Arbol Trie todos los números binarios correspondientes entre 0 y 100. Cada vez que se ingresen valores para realizar cualquiera de las dos operaciones, se los deberá buscar en el árbol, calcular y devolver el resultado. El árbol maneja sólo números binarios, en cambio el usuario debe manejar sólo números en base 10. Nota: se puede utilizar el símbolo de terminación ‘/’.
6. Un algoritmo de compresión dado se basa en la repetición de secuencias que suelen existir en los archivos. Si se encuentra una secuencia larga que apareció con anterioridad, se sustituye la secuencia por una referencia *COPIA(X, Y)*, que indica que en ese lugar se deben colocar *X* caracteres empezando por los que aparecieron *Y* posiciones antes en ese mismo archivo. Por ejemplo, la cadena: “Modulador-Demodulador...” se comprimiría como: “Modulador-DemCOPIA(8, 12)...”. Utilizando Tries se facilitaría la búsqueda de secuencias que han aparecido con anterioridad. Hacer un programa que a partir de un archivo comprima los datos y los guarde en otro archivo; probar el proceso inverso.
7. En un árbol Trie queremos representar palabras acentuadas, por lo que se propone añadir el conjunto de caracteres los siguientes (Á, á, É, é, Í, í, Ó, ó, Ú, ú). Explicar cómo afectaría a la memoria y al tiempo de ejecución, con las representaciones de arreglos enlazados y con listas enlazadas. ¿Qué otras alternativas existen para representar palabras con acento?.
8. En un Grafo sin ciclos se encuentran almacenadas todas las provincias de la Argentina con sus respectivas distancias (tomando como punto de referencia la capital de la misma). Cada provincia cuenta con sus habitantes que se encuentran almacenados en un Arbol Trie correspondiente, con una referencia desde el nodo del grafo a dicho árbol. Se pide un programa que realice las siguientes operaciones:
  - a) Cargar algunos datos para probarlo.
  - b) Consultar si se encuentra o no un habitante en una provincia determinada.

**TRABAJO PRACTICO N°4**  
**Tema: Arboles Tries**

---

- c) Un habitante de una provincia X quiere saber a que distancia está de un habitante en la provincia Y. Para ello, deberá usar el método del inciso b y luego, a partir de la ciudad que éste devuelva (Y), obtener la distancia de X a Y.
- d) Comparar dos provincias y decir cuál tiene mas habitantes.
- e) Mostrar los habitantes de una provincia determinada.

Nota: los habitantes están guardados según apellido y nombre. Asumimos que no hay personas con mismo apellido y nombre.