Tries

Del inglés reTRIEval. Se prefiere la pronunciación “try” para distinguirlo de la estructura “tree”.

Tries también conocidos como prefix trees son árboles de búsqueda. A diferencia con los árboles binarios vistos hasta ahora los nodos no almacenan la llave completa sino que sólo almacenan un símbolo de la secuencia de la llave. Las **llaves** tienen que ser **cadenas de símbolos**. La posición de un nodo en el árbol (comenzando desde la raíz) corresponde a la posición del símbolo en la llave almacenada. La llave indica cómo moverte por el árbol para llegar al dato deseado (en muchas aplicaciones el dato deseado es la llave misma).

Se usan en especial para guardar cadenas. Es un árbol de búsqueda en donde las búsquedas van a ser rápidas y compactas. Son árboles de 'prefijo'.

**INSERTAR:**

Insertar no depende de cuántas 'palabras' hayamos ingresado pero más bien de la longitud del elemento de lo que queremos ingresar.

Cuando terminamos una palabra agregamos un nodo al final que sea True. El nodo no tiene que estar vacío para significar el final de una palabra. Si el nodo de S de 'presto' estuviera verdadero significa que también tendríamos las palabra 'pre'.

En la raíz va el primer símbolo de las palabras.

Cuando un nodo es true significa que todos los nodos anteriores son una palabra (sin incluir el nodo que está en True).

Para agregar:

* ¿Está la primera letra?
  + No, creas el nodo con esa letra
  + Si, te vas por ese camino y repites hasta checar todos los símbolos (letras) de la cadena

|  | Tenemos Presto y ama |
| --- | --- |
|  | Antes: Presto y ama  Agregamos: amazon y amlo |

**BORRAR:**

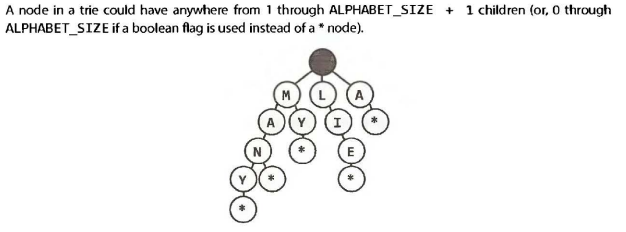
|  | Para borrar una palabra cambiamos el nodo que viene después (la que indica que es una palabra) de verdadero a falso. Si el nodo vacío ya está en false, te vas al nodo anterior y lo borras si es que es false. Dejas de subir y borrar hasta que uno de los nodos tenga otro elemento. |
| --- | --- |

:)

Un trie es un árbol k-nario en el cual los caracteres son almacenados en cada nodo. Cada camino en un árbol representa una palabra.

Se usan nodos “**\***” para decir que una palabra está completa.

Una tabla hash puede buscar palabras de manera muy rápida, pero sólo palabras completas. Un trie nos permite saber si un substring es un prefijo válido de una palabra completa.



Fuente:

McDowell, G., 2015. *Cracking the coding interview*. 6th ed. Palo Alto, California: CareerCup, p.105.