

Tries

Algoritmos y Estructuras de Datos
Segundo cuatrimestre de 2023

(2) Cómo harían el caso de las búsquedas?

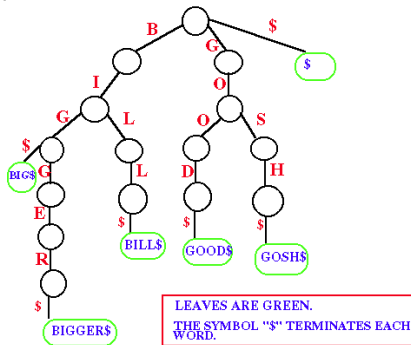
- El caso es el que uno tipeando y aparecen las palabras que coinciden...
- Con un ABB? Con un AVL? Con una tabla de hash?
- ¿Qué complejidad tienen esas búsquedas?
- ¿Pero cuánto tarda buscar en strings?
- Hoy vamos a ver una estructura que sirve para esos escenarios, además de permitir otras varias aplicaciones.

(3) Suposiciones

- Esta estructura de datos requiere concentrarse en las claves.
- Vamos a suponer que las claves x pertenecen a un alfabeto Σ^* .
- Tenemos que pensar que no tienen todas tamaño constante ($O(1)$ para comparar), sino que tienen una longitud $|x|$.
- Esa longitud puede ser:
 - La longitud propiamente dicha, en caso de strings.
 - $O(\log(x))$ si x es un entero. ¿se ve por qué?
- Además, definimos a $k = |\Sigma|$, como la cantidad de símbolos distintos del alfabeto Σ .
- ¿ Se les ocurre alguna representación para esto ? Nos suelen gustar los árboles 😊
- Pensemos en árboles k -arios.

(4) Tries

Representemos {BIG, BIGGER, BILL, GOOD, GOSH}.

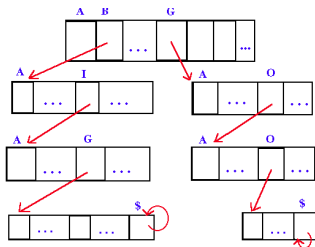


Debidos a Edward Fredkin, años '60. El nombre proviene de "retrieval".

(5) Análisis

- Complejidad de insertar clave x ? $O(|x|)$
- Y de buscar? $O(|x|)$.
- Notar: es independiente de la cantidad de elementos almacenados.
- Algoritmos: muy simples.
- Almacenamiento: punto flojo.

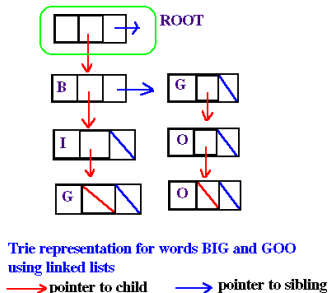
(6) Representación con arreglos y punteros



- Notar: puede ser muy ineficiente en términos de espacio.
- En especial cuando $|\Sigma|$ es grande y n es chico.
- ¿Podemos “comprimir” el arreglo? ¿Y qué pasa con la complejidad?
- En el peor caso se multiplica por k .
- También los van a ver como rose trees
- `data Tree a = Leaf a | Node [Tree a]`

(7) Tries: Representación con listas de arreglos

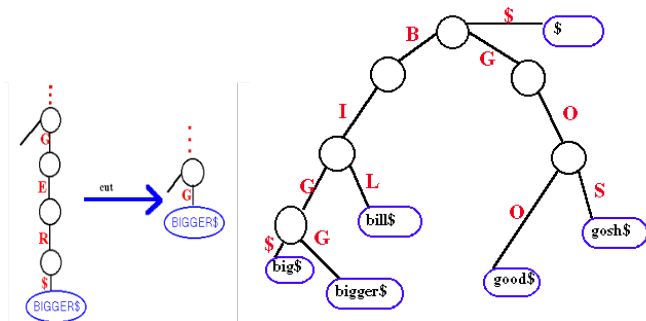
Representar a los hijos de un nodo como una lista encadenada



- Eficiente en términos de tiempo sólo si hay pocas claves
- Requiere algoritmos sobre listas
- Mucho más eficiente en términos de memoria
- Mucho menos utilizado

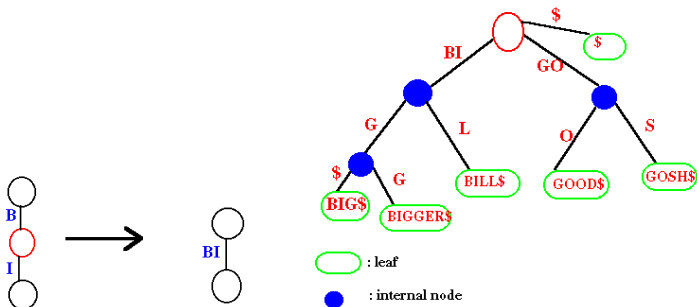
(8) Compactos

¿Qué pasa si compactamos las cadenas que sólo llevan a hojas?



(9) PATRICIA

- PATRICIA = *Practical Algorithm To Retrieve Information Coded In Alphanumeric*
- Debidos a D. R. Morrison.
- Ahora colapsamos todas las cadenas.
- Un eje puede representar más de un caracter.



(10) PATRICIA II

- La altura de un árbol Patricia binario está acotada por n (el número de claves).
- Eso no sucede con los modelos anteriores.
- En realidad, Patricia es un poco más elaborado, y hay numerosas variantes de los métodos vistos hoy.

(11) Vimos hoy

Vimos hoy:

- Tries: Árboles para recuperación de información.
- Mejoras colapsando ramas que sólo llevan a hojas.
- PATRICIA: Una mejora adicional a este tipo de árboles.

Lo que viene:

- Hashing.