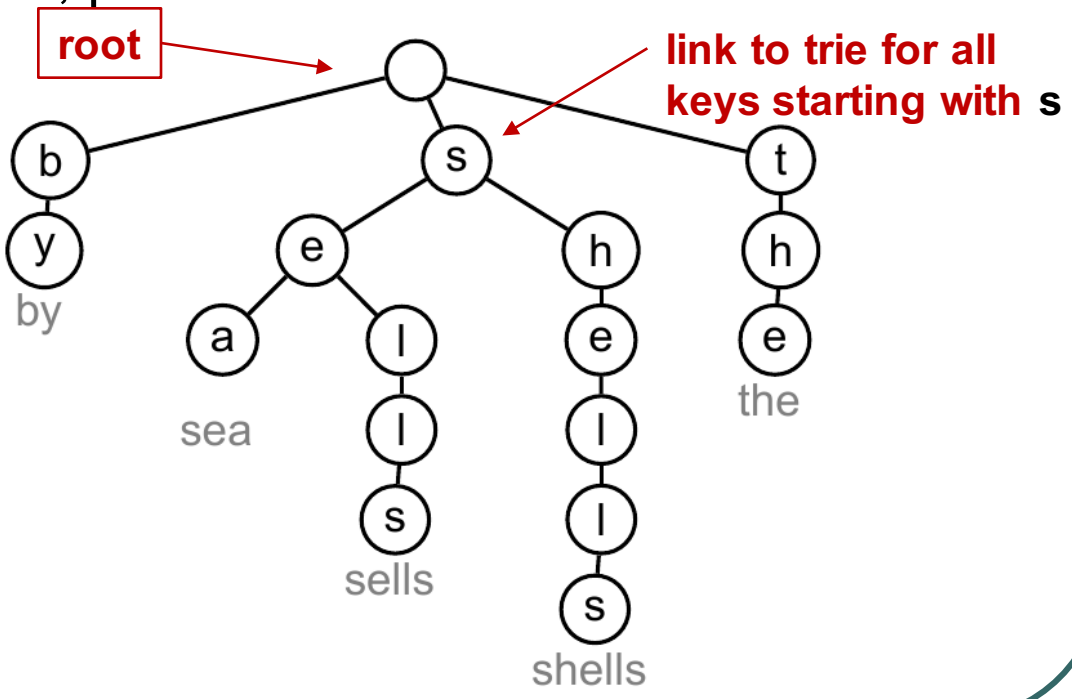
Tries

Tries

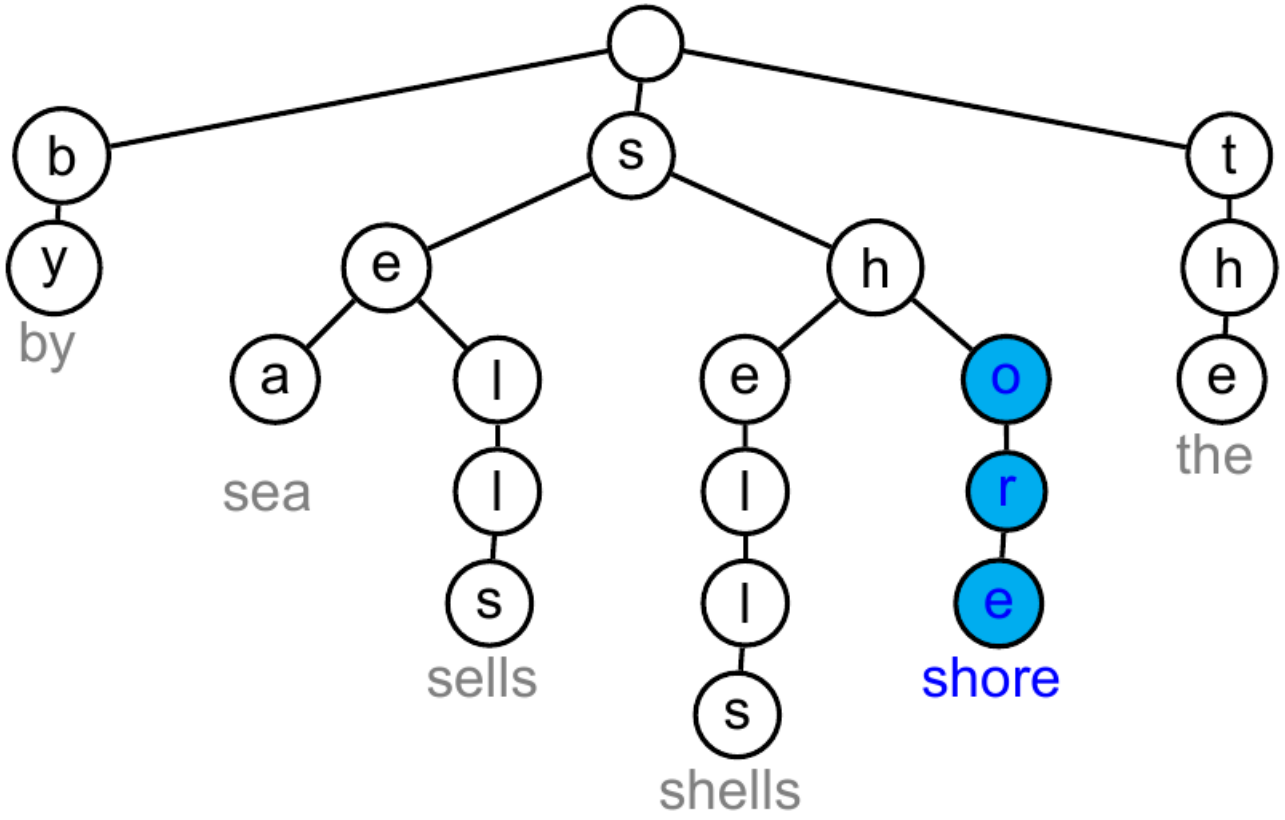
- cautare pe siruri (strings)
- stocheaza caractere in nodurile interne, nu cheile intregi (i.e. tot sirul)
- inregistrarile se stocheaza in nodurile externe (e.g. definitia cuvintelor)
- fiecare nod *R* copii (*R-way tries*)
- relatie de ordine pe caractere, pentru cautare
- Ex. sells sea shells by the sea



Tries

- Ex. sells sea

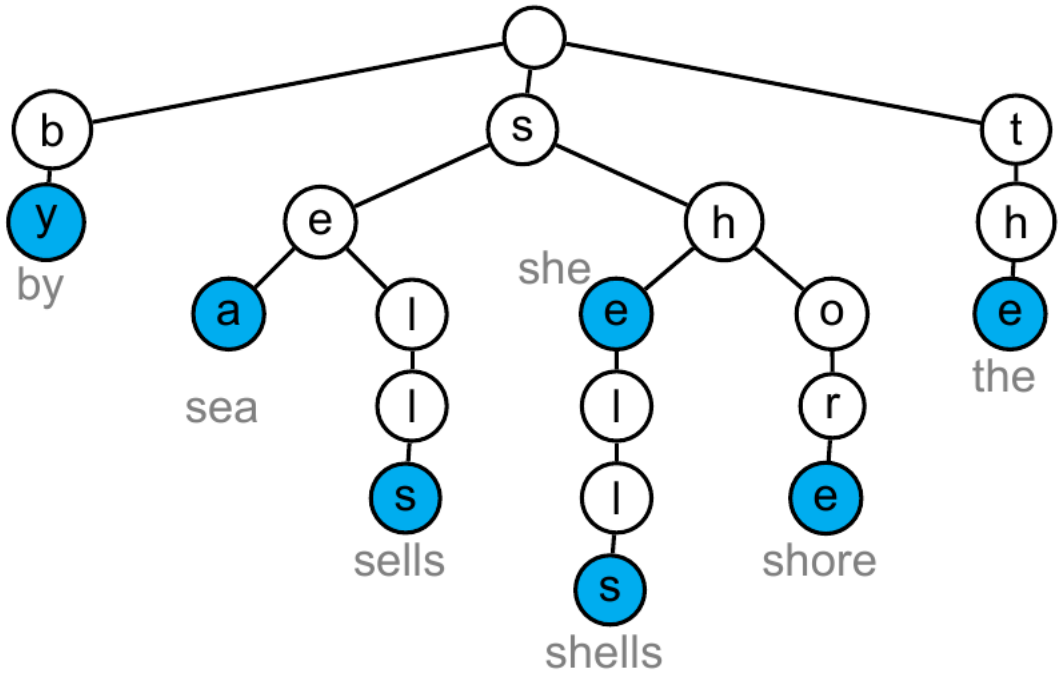
shells by the sea
shore



Tries

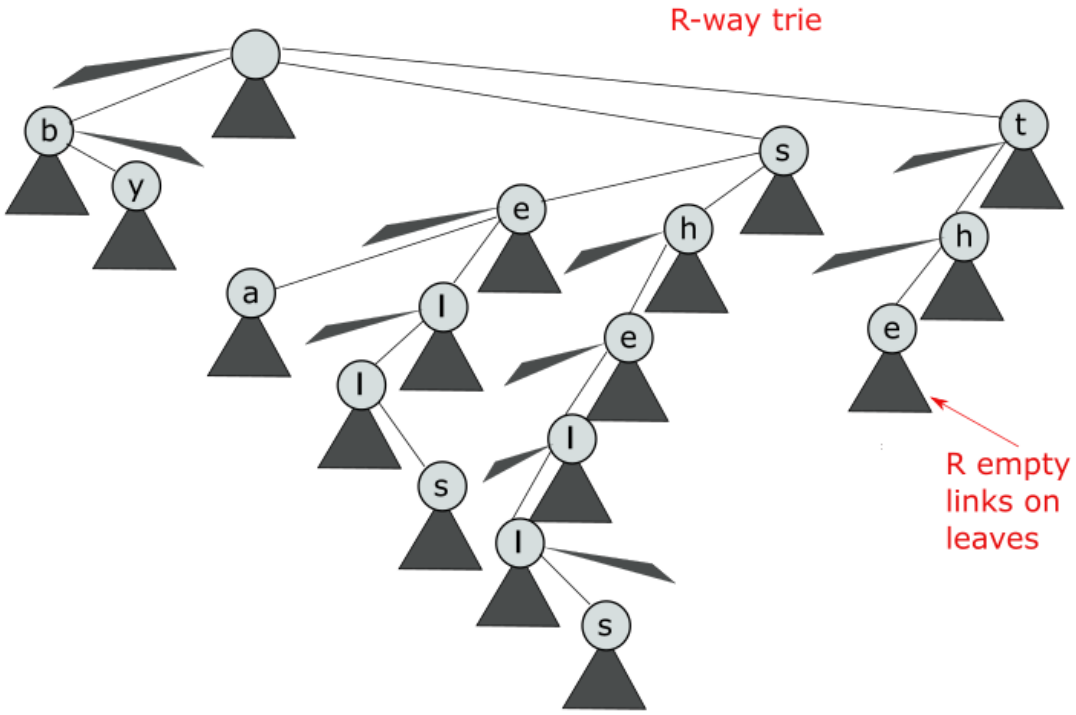
- Q: Cum tratam cazurile in care o cheie este prefix al altei chei?
  - A1: concatenam caracterul santinela ‘\0’ fiecarei chei, astfel incat sa nu se intample asta
  - A2: stocam un bit extra care sa indice care noduri corespund cheilor

Ex.
she sells sea shells by the sea shore



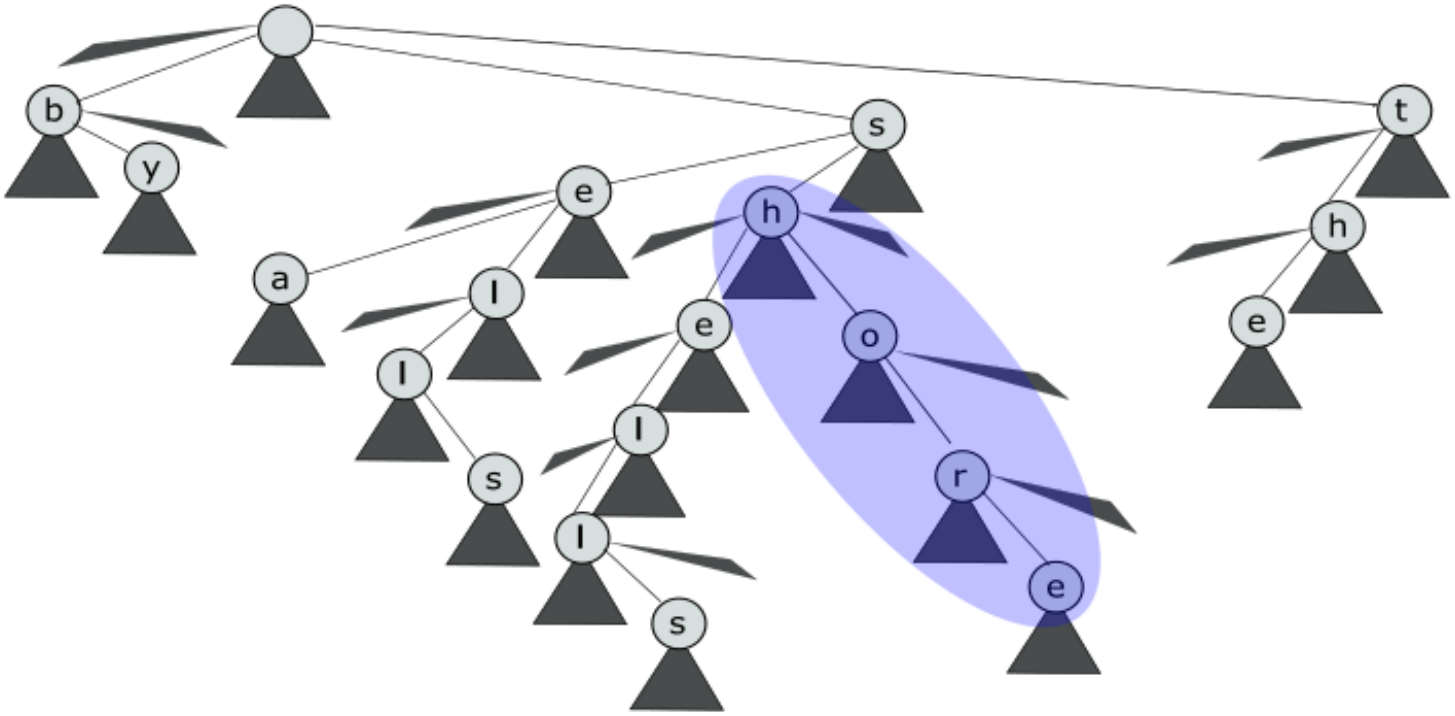
Tries - Branching

- Q: Cum se genereaza un nivel nou?
  - A: Cate o legatura pentru fiecare caracter posibil
- E.g. sells sea shells by the sea



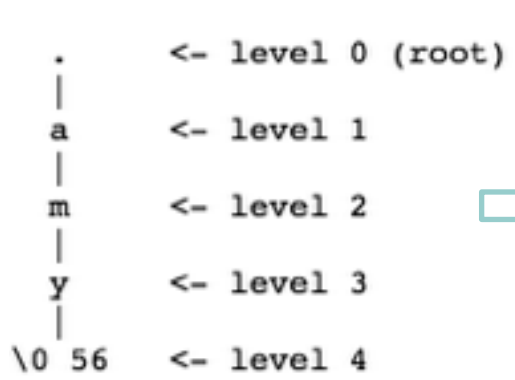
Tries - Branching

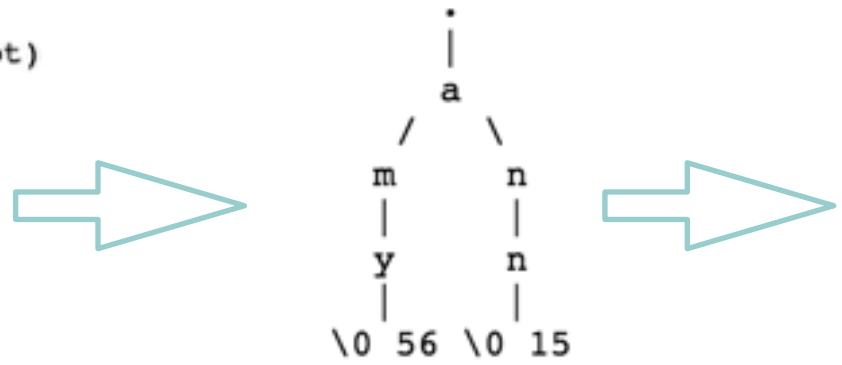
- E.g. sells sea shells by the sea shore

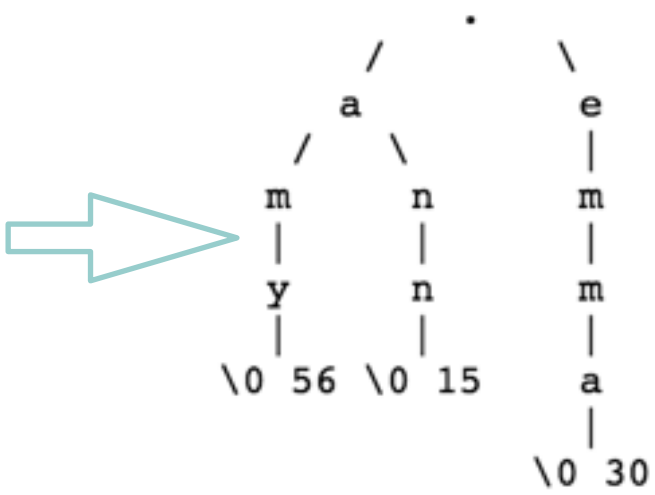


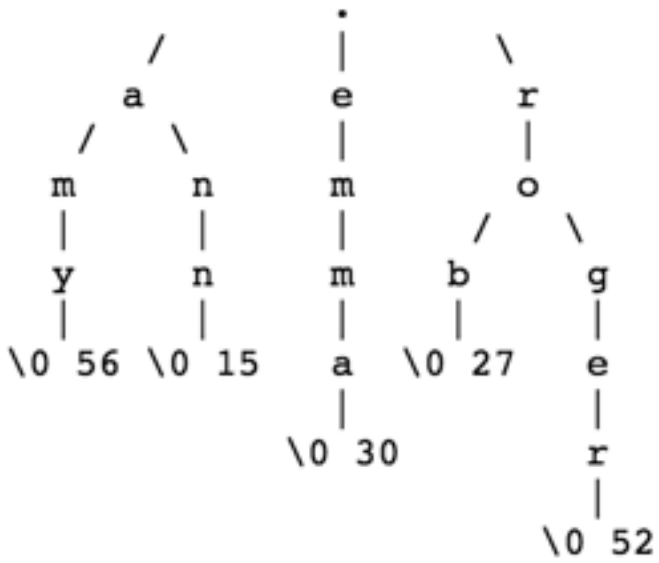
R-way Trie. Exemplu implementare

- amy 56
- ann 15
- emma 30
- rob 27
- roger 52



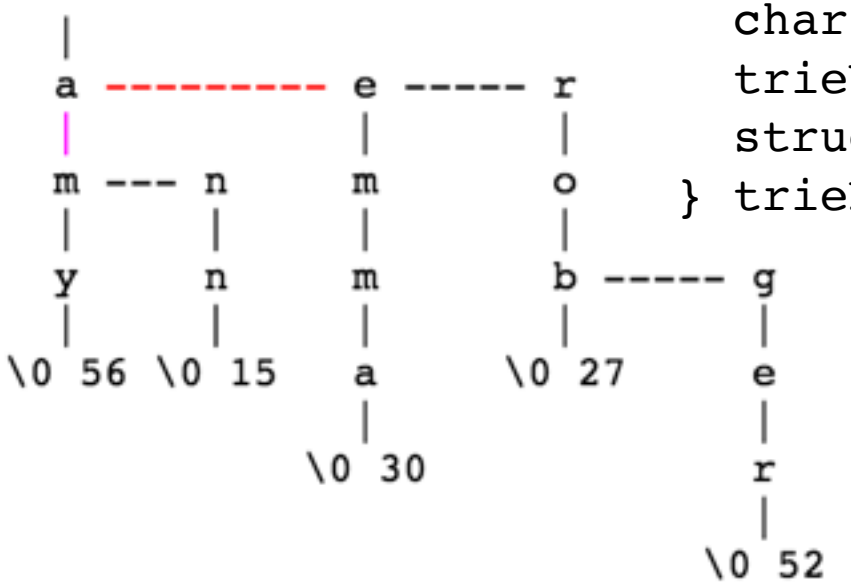






R-way Trie. Implementare

1. Cum va arata structura dupa ce adaugam **anne** cu varsta **67**? Dar **ro** cu varsta **23**?
2. Cum ar fi aratat structura daca am fi considerat o ordine de inserare diferita. e.g.: **rob**, **ann**, **emma**, **roger**, **amy**?
3. Cati copii va avea un nod? (maxim)




```
typedef struct trieNodeTag {
  char key;
  trieValueT value;
  struct trieNodeTag *next, *children;
} trieNodeT;
```

Tries - Cautare

```
IsMember(trie, key) [iterative]

1. Search top level for node that
   matches first character in key
2. If none,
     return false
   Else,
3. If the matched character is '\0'?
     return true
   Else,
4.   Move to sub-trie that matched this character
5.   Advance to next character in key*
6.   Repeat step 1

* I.e., the new search key becomes the old one without its first character.
```

Tema - inserarea

Eficienta?

Bibliografie

https://www.cs.bu.edu/teaching/c/tree/trie/