TRIE

Discuții Examen

- Poate facem un Q&A ? Cam când ați vrea ?
- Puteți pune întrebări aici și, eventual, eu voi răspunde la ele
  - Aveți voie cu materiale scrise (imi pare rau pt natura)! Mobilele pe catedra! Daca aveti mobil la voi dupa ce incepe examenul -> frauda -> restanta & posibila exmatriculare!

Subiecte Examen

- Exemplificare cum funcționează structuri de date/algoritmi & Complexitati
  - Count Sort, radix sort, quick sort, merge sort
  - Cozi, Stive, Deque
  - Hashuri
    - Inserare/Cautare/Stergere
    - Tratarea coliziunilor: Inlantuire/Adresare Directa
    - Functii de dispersie: metoda diviziunii/metoda multiplicarii
    - Rabin Karp
  - Heapuri, Heapuri Binomiale, Heapuri Fibonacci

Subiecte Examen

- Exemplificare cum funcționează structuri de date/algoritmi & Complexități
  - Arbori binari de căutare
    - Inserare, Ștergere, Căutare, Succesor, Predecesor, k-th element
    - Parcugeri Preordine, Inordine, Postordine
  - Arbori binari de căutare echilibrati:
    - inserare, ștergere, căutare, succesor, k-lea cel mai mare…
    - La alegere din
      - AVL/Red Black/**Skip lists**/B-arbori/(Treaps -> doar inserare)
  - Arbori de Intervale/ Batog
    - Inserare/Căutare min/Ștegere/Sortare/ Calculare sumă pe interval/Update pe interval
  - RMQ&LCA&LA
    - Ce rezolvă? Cum funcționează pe un exemplu? Complexitate?
  - Trie

Subiecte Examen

- Desenați un arbore binar complet de înălțime 2
- Desenați un heap cu 5 noduri
- Desenați un arbore binar de căutare cu 6 noduri. Ce înălțimi poate să aibă?
- Inserați, pe rând, într-un heap Fibonacci de minim valorile 1, 2, 9, 5, 7, 3
- Cum folosim un arbore de intervale ca să sortăm un vector?
- Se dă un arbore. Care este LA între 3 și 9? Dar 2 și 8? Cum se calculează? Ce complexitate are?
- Cum găsim succesorul într-un arbore binar de căutare?
- Construiți un TRIE cu cuvintele : ala, bala, portocala
- Bonus:
  - Demonstrați că orice algoritm care construiește un arbore binar de căutare cu n numere rulează în timp $\Omega(n \log n)$.

Subiecte Examen

- Probleme ca la seminar
  - Va trebui să scrieți cum o rezolvați și ce complexitate are soluția voastră:
  - Gen: Se dau n numere. Câte perechi de numere au suma un pătrat perfect?

Subiecte Examen

- Gen: Se dau n numere. Câte perechi de numere au suma un pătrat perfect?
  - Iau toate numerele de la 1 la max, le calculez pătratul, apoi iau toate perechile de la 1 la n, le fac suma și văd dacă dă fix pătratul la care sunt
    - Nota 2,
    - Dacă adaug și complexitate corect ??
      - O(n^2*max) -> nota 3
  - Iau toate numere de la 1 la sqrt max * 2 ….. si la fel -> nota 3, respectiv 4
    - O(n^2* sqrt(max)
  - Iau toate perechile de numere, le fac suma și văd dacă rezultatul e un pătrat perfect în O(1) (gen sqrt(x) * sqrt(x) == x)... ->5 cu complexitate O(n^2) 7
  - Iau toate numerele și toate pătratele <= max1 + max2 și văd dacă Patrat-nr există între numerele mele cu hash-uri O(n*sqrt(x)) ->5 cu complexitate O(n*sqrt(max)) 7
  - Împreună 10…

Trie

- Am mai multe cuvinte pe care le tin minte și apoi am întrebări de genul:
  - este cuvântul dat în acea listă sau nu?
- Cum putem rezolva?
  - Hash-uri!
  - Cât mă costă un query?
    - O(l), unde l e lungimea cuvântului
  - Câtă memorie mă costă să reține hash-ul?
    - O(n*l)
  - Ce credeți că am putea optimiza?
    - Memoria (poate)
    - Timpul pentru query-uri nereușite … oarecum

Trie

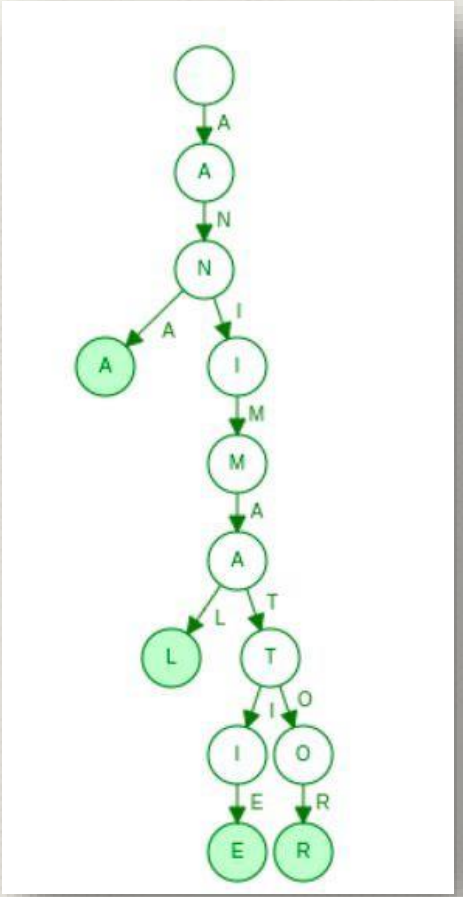
- Am mai multe cuvinte și apoi am întrebări de genul:
  - este cuvântul în dicționar sau nu?
  - care este cel mai lung prefix al cuvântului în dicționar?
- Mai merge cu hash-uri ?
  - Nu prea …
- Alte soluții?
  - Sortăm toate cuvintele lexicografic și apoi căutăm binar
  - Ținem toate cuvintele într-un arbore binar de căutare echilibrat
- Ambele soluții au O(n*l) memorie și **O(logn *l)** complexitate pe search
- Arborele binar permite, totuși, și inserări și ștergeri!!

Trie

- Dacă avem cuvintele **anima, animal, animație, animator, animare,** reținem, pentru fiecare, prefixul **anim** comun.
- Cum credeți că putem îmbunătăți memoria folosită?
  - Am putea, când le ținem sortate, să le ținem ceva de genul:
    - anima
    - 5l
    - 5tie
  - Adică, să ținem lungimea prefixului față de elementul anterior
  - Putem duce o idee similară și spre arbori binari de căutare, dar să nu ne mai complicăm :)

Trie

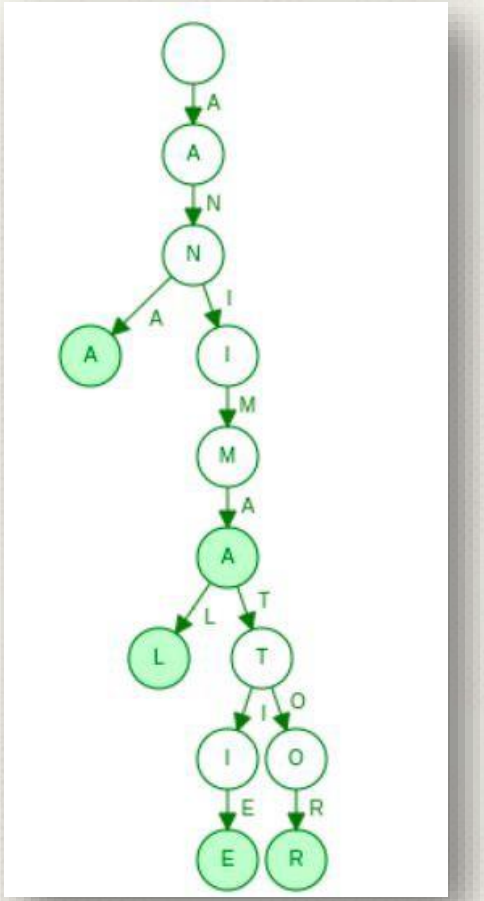
Trie cu cuvintele **ana, animator, animație, animal**



Trie

Trie cu cuvintele **ana, animator, animație, animal, anima.**

vizualizare trie



Trie - Memorare

- Cum îl reținem?
  - Fiecare nod are un vector cu 26 de vecini, una pentru fiecare literă (sau mărimea alfabetului)
  - Ce facem dacă alfabetul e mare?
  - Fiecare nod ține un hash_map care, pentru fiecare literă, ține pointer-ul către nodul cu acea literă.

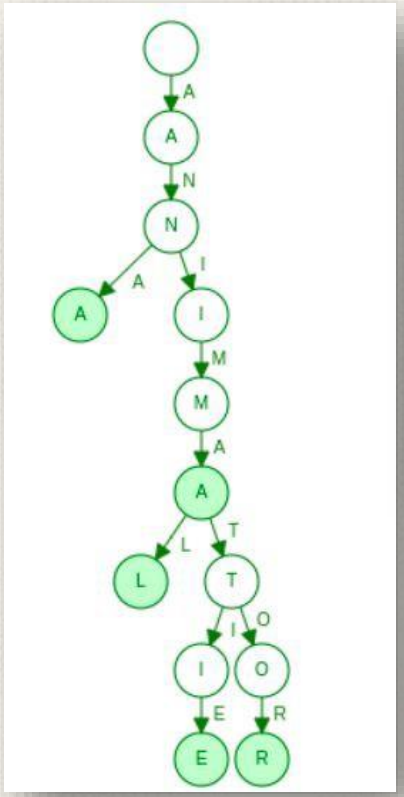
Trie - Inserare

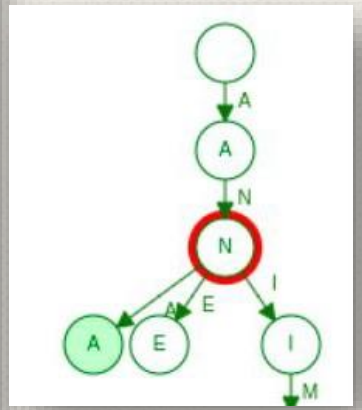
Pornim din rădăcină și, la fiecare literă, mergem în nodul corespunzător literei. Eventual creăm acel nod.

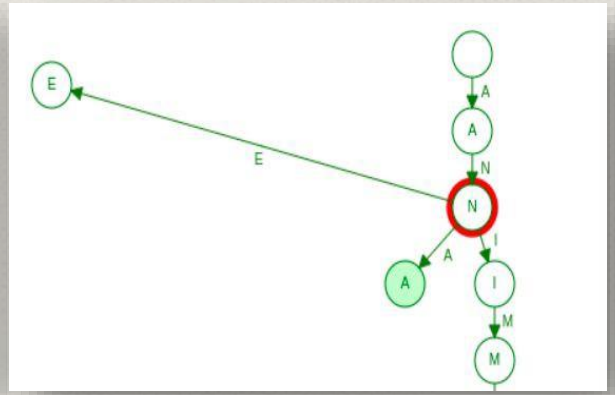
https://www.cs.usfca.edu/~galles/visualization/Trie.html

Trie - Inserare

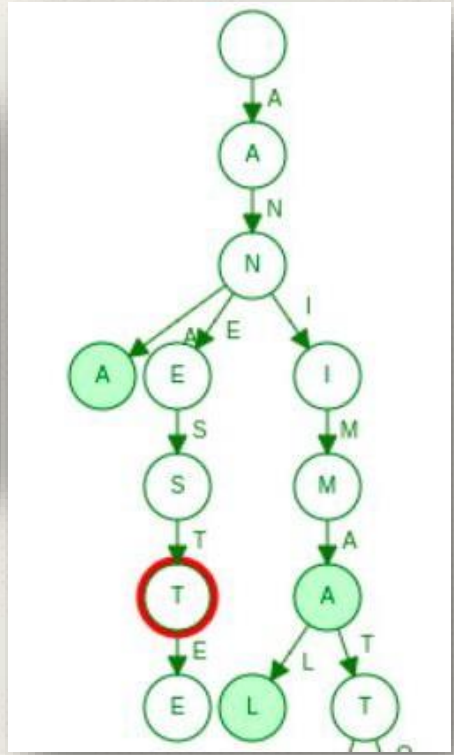
Inserăm **anestezie**







Complexitate: O(l)



Trie - Inserare

Complexitate: O(l)

Trie - Căutare

Pornim din rădăcină și mergem, la fiecare pas, pe litera corespunzătoare.

Complexitate O(l) pentru căutare reușită.

În practică, mai rapid pentru căutare nereușită.

Căutare prefix maxim:

- Căutăm elementul până nu găsim nod corespunzător acelei litere

Kahoot

Succes în sesiune :)