

TRIE





Kahoot



Discuții Examen

- Ca de obicei, mă găsiți online, unde răspund la întrebări
- Poate facem un Q&A ? Cam cand ati vrea ? Nu promit nimic inca... Incercam pe 8
- Puteți pune întrebări aici și eventual eu voi răspunde la ele

Subiecte Examen

- 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!
- Doua parti:
 - Exemplificare cum funcționează structuri de date/algoritmi
 - Probleme ca la seminar
 - Pauza 15 minute între ele

Subiecte Examen

- Exemplificare cum funcționează structuri de date/algoritmi & Complexitati
 - Count Sort, radix sort, quick sort, merge sort
 - Cozi, Stive, Deque
 - Hashuri
 - Inserare/Cutare/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 & Complexitati
 - Arbori binari de cautare
 - Inserare, Stergere, Cautare, Succesor, Predesor, k-th element
 - Parcugerii Preordine, Inordine, Postordine
 - Arbori binari de cautare echilibrati:
 - inserare, stergere, cautare, succesor, k-lea cel mai mare...
 - La alegere din
 - AVL/Red Black/Skip lists/B-arbori/Treaps
 - Arbori de Intervale/ Batog
 - Inserare/Cautare min/Stegere/Sortare/ Calcularea suma pe interval/Update pe interval
 - RMQ&LCA&LA
 - ce rezolva, cum functioneaza pe un exemplu, complexitate..
 - Trie

Subiecte Examen

- Desenati un arbore binar complet de inaltime 2
- Desenati un heap cu 5 noduri
- Desenati un arbore binar de cautare cu 6 noduri. Ce inaltime poate sa aiba?
- Inserati pe rand intr-un heap fibonacii de minim valorile 1, 2, 9, 5, 7, 3
- Cum folosim un arbore de intervale sa sortam un vector?
- Se da un arbore, care este LA intre 3 si 9, dar 2 si 8, cum se calculeaza ? Ce complexitate are ?
- Cum gasim succesorul intr-un arbore binar de cautare
- Construiti un TRIE cu cuvintele : ala, bala, portocala
- Bonus:
 - Demonstrati ca orice algoritm care construiește un arbore binar de cautare cu n numere ruleaza in timp $\Omega(n \log n)$.

Subiecte Examen

- Probleme ca la seminar
 - 3-4 probleme in o ora (o sa fac un mic test la ultimul seminar)
 - O sa va dau la fiecare grupa o problema in 15-20 minute la seminarul 7 si o sa incerc sa va dau feedback ulterior... Daca o rezolvati bine -> puncte activitate...
 - Va trebui sa scrieti cum o rezolvati si ce complexitate are solutia voastra:
 - Gen: Se dau n numere, cate perechi de numere au suma un patrat perfect
 -
 -

Subiecte Examen

- Gen: Se dau n numere, cate perechi de numere au suma un patrat perfect
 - Iau toate numerele de la 1 la \max le calculez patratul, apoi iau toate perechile de la 1 la n si le fac suma si vad daca da fix patratul la care sunt
 - Nota 2,
 - Daca adaug si 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 si vad daca rezultatul e un patrat perfect in $O(1)$ (gen $\sqrt{x} * \sqrt{x} == x$)... ->5 cu complexitate $O(n^2)$ 7
 - Iau toate numerele si toate patratele $\leq \max1 + \max2$ si vad daca Patrat-nr exista intre numerele mele cu hashuri $O(n * \sqrt{x})$ ->5 cu complexitate $O(n * \sqrt{\max})$ 7
 - Impreuna 10...

Trie

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

Trie

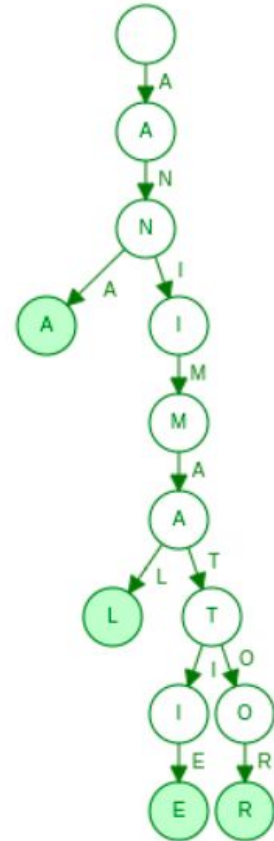
- Am mai multe cuvinte și apoi am întrebări de genul:
 - este cuvântul în dicționar sau nu?
 - care este cel mai lung prefix al cuvântului în dicționar?
- Mai merge cu hash-uri ?
 - Nu prea ...
- Alte soluții?
 - Sortăm toate cuvintele lexicografic și apoi căutăm binar
 - Ținem toate cuvintele într-un arbore binar de căutare echilibrat
- Ambele soluții au $O(n \cdot l)$ memorie și $O(\log n \cdot l)$ 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

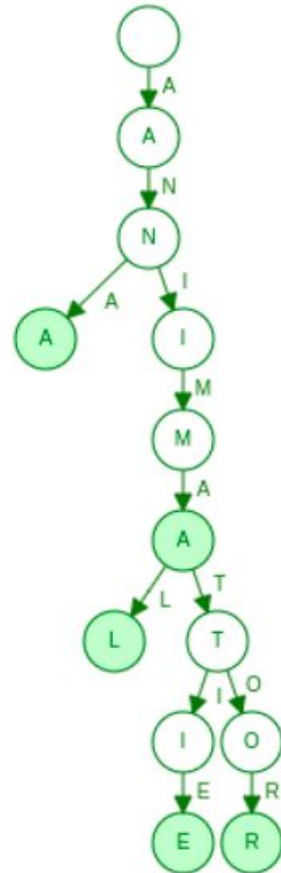
Trie cu cuvintele **ana**, **animator**, **animație**, **animal**



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 tine pointerul catre nodul cu acea litera

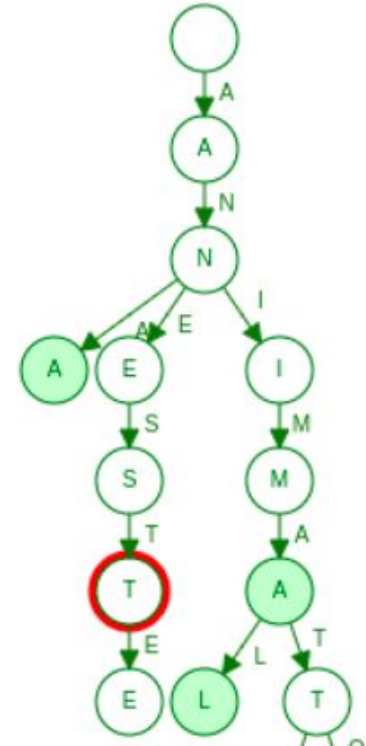
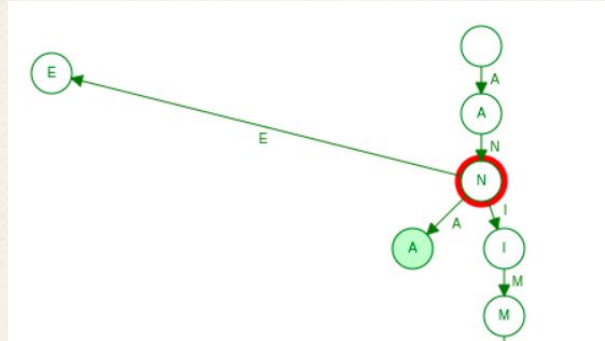
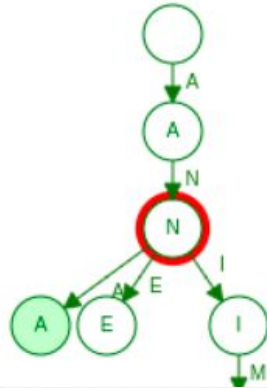
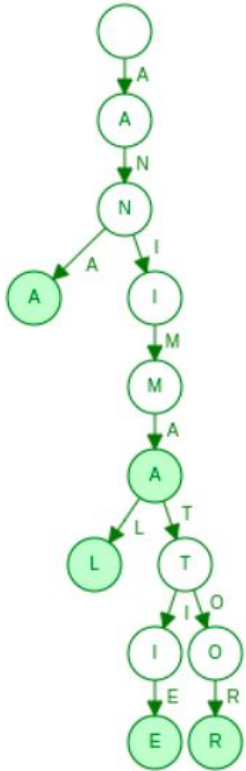
Trie - Inserare

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



Trie - Inserare

Complexitate: $O(1)$

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

Succes în sesiune :)