

Tema 3

- Structuri de Date -

Informatii trimitere:

- Tema se va incarca atat pe [vmchecker](http://vmchecker.com) cat si pe cs.curs.pub.ro
- **Soft deadline:** 17 mai, 2017
- **Hard deadline:** 20 mai, 2017
- Pentru fiecare zi de întârziere se va scădea câte 1 punct din 10

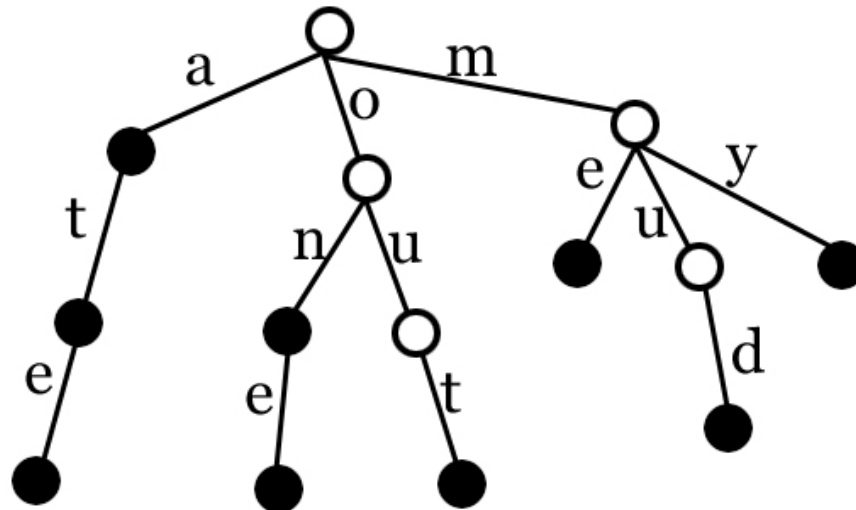
1. Abstract

Trie-ul este o structura de date sub forma unui arbore de cautare, folosita deseori pentru a efectua anumite operatii asupra unor seturi de cuvinte. Acesta are o structura oarecum asemanatoare cu a unui arbore binar de cautare, cu cateva diferente notabile:

- Fiecare nod din trie poate avea de la 0 la $|\Sigma|$ fii, unde $|\Sigma|$ reprezinta dimensiunea alfabetului.
- Spre deosebire de arborii de cautare, unde fiecarui nod i se asocia o cheie, la trie cheile se asociaza caii pana la nod
- De interes pentru noi sunt, in principal, acele noduri in care 'se termina' cuvinte
- Radacina trie-ului corespunde intotdeauna cuvintului vid

Mai jos avem reprezentarea unui astfel de trie:

- Cercurile reprezinta noduri din arbore
- Cercurile pline reprezinta noduri in care se termina cuvinte (ex: a, at, ate, on, one, out etc.)
- Cercurile goale reprezinta noduri in care nu se termina niciun cuvânt
- Dupa cum se vede, cheile nu se stocheaza in noduri, ci se compun pe masura ce se avanseaza in adancime



2. Cerinta

Scopul temei este implementarea unui astfel de arbore care sa suporte urmatoarele operatii:

1. add w
2. remove w
3. find w
4. find_longest_prefix w
5. find_all_with_prefix [pref]
6. mean_length [pref]

Pentru simplitate, in rezolvarea temei puteti considera ca toate cuvintele sunt alcatuite din caractere lowercase si cifre ([a-z][0-9]).

2.1 Add w

Dupa cum sugereaza si numele, operatia adauga cuvantul **w** in trie.

2.2 Remove w

Operatia scoate cuvantul **w** din trie.

Atentie! Eliminarea unui cuvnt din trie va avea drept consecinta stergerea unor noduri doar atunci cand acestea se afla pe o cale pe care nu se mai gaseste niciun alt cuvnt. De exemplu, eliminarea cuvntului 'mud' din trie-ul de mai sus ar duce la stergerea a 2 noduri. Pe de alta parte, eliminarea cuvntului 'at' nu ar produce stergerea niciunui nod.

Daca cuvantul **w** nu exista in trie, operatia nu face nimic.

Hint: Pe baza exemplului dat, ar fi indicat sa gasiti o modalitate prin care sa diferentiati nodurile in care se termina un cuvnt de cele in care nu se termina

niciun cuvânt și să actualizați această proprietate a nodurilor după fiecare add sau remove.

2.3 Find w

Căuta cuvântul **w** în trie și printează True dacă acesta a fost găsit, respectiv False dacă nu există în arbore.

2.4 Find_longest_prefix w

Această operație returnează cel mai lung prefix pe care cuvântul **w** îl are în comun cu orice alt cuvânt din trie.

Dacă cuvântul **w** nu are un prefix comun cu niciun alt cuvânt din trie, se returnează 'None'.

Pe trie-ul dat ca exemplu, 'find_longest_prefix one' ar avea ca rezultat 'one', în timp ce 'find_longest_prefix quagmire' ar avea ca rezultat 'None'.

Atentie! Nu este obligatoriu ca **w** să existe în trie. De exemplu, operația 'find_longest_prefix our' trebuie să returneze 'ou'.

2.5 Find_all_with_prefix [pref]

Operația listează, în ordine alfabetică, pe o singură linie, toate cuvintele care încep cu prefixul **pref**, separate între ele printr-un spațiu.

Dacă nu există niciun cuvânt care începe cu **pref**, atunci se va afișa 'None'.

Dacă find_all_with_prefix nu este însoțită de niciun prefix (adică pref = ""), atunci se vor lista toate cuvintele din trie.

2.6 Mean_length [pref]

Calculează lungimea medie a cuvintelor din trie care încep cu un prefixul **pref** (media va fi afișată cu 3 zecimale).

De exemplu, pe arborele de mai sus, 'mean_lenght m' va produce $(2+2+3)/3=2.333$

Dacă nu există niciun cuvânt care începe cu **pref**, atunci operația va returna 0.

Ca și mai sus, dacă operația nu este însoțită de niciun prefix, se va calcula lungimea medie pentru toate cuvintele din trie.

3. Implementare

3.1 Rulare

Programul va fi rulat astfel:

```
./tema3 input.in output.out
```

Unde:

input.in -> numele fisierului de intrare

output.out -> numele fisierului de iesire

3.2 Formatul datelor de intrare/iesire

Pe prima linie a fisierului de intrare se va afla un numar **N** care va indica numarul de comenzi continute in fisier. Pe urmatoarele **N** linii ale fisierului se vor gasi comenzi (cate o comanda pe linie).

3.3 Exemplu

Un exemplu de rulare este urmatorul:

| input.in | output.out |
|--|--------------------------|
| 1 13 | ana |
| 2 add ana | False |
| 3 add owen | owe |
| 4 add anastasia | 0 |
| 5 add ouroboros | 7.333 |
| 6 find_longest_prefix anagram | our |
| 7 remove ana | ouroboros owen |
| 8 find ana | anastasia ouroboros owen |
| 9 find_longest_prefix owe | |
| 10 mean_length john | |
| 11 mean_length | |
| 12 find_longest_prefix ourlordandsaviorcthulhu | |
| 13 find_all_with_prefix o | |
| 14 find_all_with_prefix | |

4. Notare

- **85 de puncte** conform testelor de pe vmchecker
- **10 puncte**: coding style, cod comentat si usor de citit
- **5 puncte** README in care explicati cum ati abordat tema, ce decizii ati luat in implementarea ei si de ce
- **20 de puncte** bonus pentru implementarile ce nu au memory leak-uri (bonusul este conditionat de trecerea tuturor testelor pe vmchecker)
- Este foarte indicata folosirea [stilului de cod din kernelul de linux](#)

Se vor depuncta:

- Alocari statice
- Lipsa README-ului

- Linii de cod mai lungi de 80 de caractere
- Warning-urile de compilare (trebuie compilat cu flag-ul -Wall)
- Verificarea codurilor de eroare
- Neeliberarea memoriei la sfarsitul executiei
- Cazuri exceptionale de cod ilizibil
- Folosirea gresita a pointerilor
- **Temele care nu compileaza sau nu ruleaza se vor nota automat cu 0p.**
- Implementarile care nu respecta abordarea de trie si/sau implementeaza functiile cerute prin alte metode se vor depuncta la latitudinea corectorului.

5. Reguli de trimitere a temelor

Temele se vor incarca atat pe vmchecker cat si pe cs.curs.pub.ro.

Arhiva cu rezolvarea temei trebuie sa contina:

- Fisiere sursa semnate cu numele si grupa voastra
- README
- Makefile care sa contina urmatoarele reguli
 - ◆ build - care compileaza sursele intr-un executabil cu numele tema3
 - ◆ clean - care sterge executabilul si toate fisierele obiect rezultate

Arhiva nu trebuie sa contina fisiere executabile sau obiect.

Orice tema a carei arhiva nu respecta formatul de mai sus nu va fi luata in considerare.