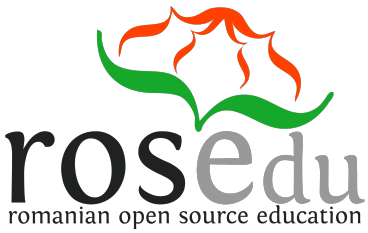

***R**omanian **O**pen **S**ource **Edu**cation*

Programming Assignment

Curs de Dezvoltare Liberă

Autor:
Tudor Cebere

Email:
tudorcebere@rosedu.org

February 14, 2020

1 Admitere

Pentru a fi admiși în cadrul programului CDL, studenții trebuie să rezolve problema propusă și să trimită soluția conform detaliilor specificate.

Doar submisiile care implementează funcționalitățile de bază descrise vor fi luate în considerare. Fiecare submisie validă va fi evaluată de către mentori, fiecare mentor alegând un set de rezolvări pe care le consideră pentru mentorat.

2 Problemă - Open Source Search Engine

2.1 Enunț

Dorin își dorește să implementeze un motor de căutare de bază în documente pentru a găsi mai ușor fișierele al căror conținut îl interesează. El știe că fișierele în care caută nu se modifică niciodată și că motorul său de căutare folosește query-uri scrise sub formă logică. Când face o interogare, el vrea să găsească toate documentele care respectă o anumită schemă logică pe baza cuvintelor conținute de document.

Exemple de query-uri:

- `(Linus || Torvalds) && kernel && C`

  Dorin vrea să găsească toate documentele care conțin string-urile "Linus" sau "Torvalds", alături de string-urile "kernel" și "C".

- `!(java) && cool && programming && languages`

  Dorin vrea să găsească toate documentele care conțin string-urile "cool", "programming" și "languages", dar nu și string-ul "java".

- `Machine && Learning && !(Siraj || Raval)`

  Dorin vrea să găsească toate documentele care conțin string-urile "Machine" și "Learning", dar nu conțin string-urile "Siraj" sau "Raval".

Observații legate de query-uri:

- Mereu vor fi parantezate corect.
- Nu pot avea mai mult de 20 de termeni.
- Pot conține cuvinte lowercase sau uppercase, dar care vor fi considerate ca lowercase în cadrul căutării. Căutarea este una case-insensitive, cuvintele "Andrei" și "andrei" fiind echivalente.

De asemenea, Dorin se întreabă cum să reprezinte datele în memorie astfel încât un lookup să fie cât mai puțin costisitor. Prima soluție la care se gândește este să rezolve schema logică cerută și să caute liniar prin fiecare document, dar această abordare nu este suficient de eficientă.

După o perioadă, Dorin se gândește la o structură de tip inverted index, în care va reține în ce document apare fiecare cuvânt, după urmatoarea schemă:

Cuvânt	doc1	doc2	doc3
open	1	0	1
source	1	0	0
kernel	0	0	1
security	0	1	0
C	1	1	1

Observăm că marcăm existența unui cuvânt într-un document prin 1 dacă există sau 0 dacă nu există.

Astfel, pentru a găsi documentele în care apar aceste cuvinte, folosim array-urile binare asociate cuvintelor și aplicăm schema logica cerută. La final, fiecare document care respectă schema logică va fi marcat cu 1.

Exemplu de rezolvare a unui query:

```
open && !(source) && (security || C)
```

```
[1 0 1] && !([1 0 0]) && ([0 1 0] | [1 1 1])
[1 0 1] && [0 1 1] && [1 1 1]
[0 0 1]
```

doc3 este singurul document care respectă toate cerințele din căutarea noastră.

Cum lui Dorin nu prea i-a plăcut cartea, te roagă pe tine să îl ajuți să implementeze acest search engine. Dacă reușești să îl ajuți pe Dorin în implementare, îți va pune o vorbă bună cu mentorii din cadrul CDL.

2.2 Funcționalități adiționale

După ce implementați soluția problemei, fiecare proiect trebuie să implementeze **minim o funcționalitate în plus** (dar cu cât mai multe, cu atât mai bine). Acestea pot fi atât din lista de mai jos, cât și unele gândite de voi. Încurajăm și apreciem soluțiile care vin cu idei originale.

Funcționalități posibile:

- GUI pentru search engine
- Îmbunătățirea search engine-ului: tf-idf
- Paralelizarea operațiilor
- Set de teste și code coverage
- Folosirea bazelor de date

2.3 Informații utile

Pentru a testa soluțiile încărcate, vom folosi documentele de aici.
Toate submisiile trebuie să fie postate pe un github public, iar link-ul către repo să fie atașat în form-ul de submisie.
Toate submisiile trebuie să fie documentate în Readme.md. Poți urmări acest tutorial despre cum să faci un Readme.md bun.
Toate submisiile trebuie să aibă un sistem de build care să instaleze toate dependințele necesare proiectului. Proiectul trebuie să ruleze pe Ubuntu 18.04.

3 Trimitere soluție și Întrebări

Submisia trebuie făcută în acest form. Pentru întrebări:

- ROSEdu Slack, pe channel-ul: cdl_primavara_2020_intrebari
- Mail