Proiectarea Algoritmilor

Tema 1 – Corectorul ortografic

1. Descrierea problemei

După ce au absolvit Facultatea de Automatică și Calculatoare în anul 1998, S. și L. au inventat un motor de căutare revoluționar. Motorul de căutare returnează cele mai relevante pagini de pe Internet care au legătură cu șirul de caractere căutat.

Problema este că utilizatorii greșesc frecvent termenii pe care doresc să îi caute și de aceea se pierde foarte mult din relevanța rezultatelor. Pentru a rezolva această problemă, cei doi absolvenți au decis să implementeze un sistem care sugerează posibile corecții. Corecțiile vor fi făcute în raport cu vocabularul limbii engleze.

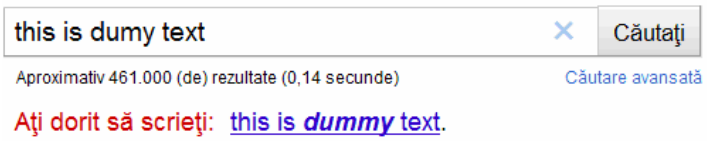

Figura 1. Atunci când cineva caută "*this is dumy text*", este sugerată o corecție.

Pentru a implementa acest sistem, S. și L. folosesc un dicționar și distanța de editare dintre două șiruri de caractere.

2. Structura dicționarului

Dicționarul folosit este unul public și se găsește pe site-ul de curs (cs.curs.pub.ro) în secțiunea de teme. Acesta este un fișier text care conține cele mai uzuale 8.000 de cuvinte ale limbii engleze. Pe fiecare linie din fișier se găsesc un șir de caractere și un număr natural, separate de exact un spațiu. Șirul reprezintă un cuvânt și va fi format din maxim 32 de litere mici ale alfabetului englez. Numărul reprezintă *frecvența de apariție* a cuvântului și va putea fi reprezentat pe 32 de biți cu semn. Cu cât frecvența este mai mare, cu atât cuvântul este mai uzual. Puteți observa de exemplu că *the* și *is* au frecvențe foarte mari, în timp ce cuvinte precum *variable* sau *procedure* au frecvențe mult mai mici. Cuvintele din dicționar sunt sortate alfabetic.

3. Distanța de editare dintre șiruri de caractere

Se definește *distanța de editare* dintre două șiruri de caractere (sau *abaterea* dintre două șiruri de caractere) ca fiind numărul minim de operații elementare care trebuiesc aplicate primului șir astfel încât să fie transformat în cel de-al doilea. Operațiile elementare sunt:

- inserarea unui caracter oriunde în șir
- ștergerea oricărui caracter din șir
- înlocuirea unui caracter cu orice alt caracter

De exemplu, distanța de editare dintre cuvintele *dummy* și *dumb* este 2: se va șterge primul *m* din *dummy*, iar *y* va fi înlocuit cu *b*. Se observă că dacă două șiruri coincid, distanța de editare dintre ele este 0.

4. Corectarea unui singur cuvânt

Atunci când se dorește corectarea unui singur cuvânt scris greșit, se pot folosi dicționarul și distanța de editare. Astfel, corecția sugerată va fi cuvântul din dicționar care are abaterea minimă față de cuvântul care trebuie corectat.

Folosind acest procedeu se pot obține însă mai multe corecții posibile, în cazul în care există cel puțin două cuvinte cu aceeași abatere minimă. De exemplu, pentru *varrry* se poate obține atât *larry* cât și *array*, deoarece ambele cuvinte au abaterea egală cu 2 față de *varrry*. În această situație, corecția indicată va fi cuvântul cu frecvența maximă. Astfel, pentru exemplul de mai sus, corecția indicată va fi *larry*. Dacă există mai multe cuvinte cu aceeși distanță minimă și frecvență maximă, atunci se va alege cuvântul care se află în dicționar pe poziția minimă (cuvântul *minim lexicografic*).

5. Corectarea unui șir de caractere

În continuare vom considera pentru simplitate că un șir de caractere este format doar din litere mici ale alfabetului englez și din spații, cuvintele fiind separate de exact un spațiu.

Atunci când trebuie corectat un șir format din mai multe cuvinte, se poate corecta independent fiecare cuvânt. Fie p_1, p_2 ... p_K cuvintele inițiale și c_1, c_2 ... c_K cuvintele identificate ca și corecții, conform punctului anterior. Se definesc:

$$\textit{Abatarea totală} \text{ a șirului} = \sum_{i=1}^{K} abatare(p_i, c_i)$$

$$\textit{Frecvența totală} \text{ a șirului} = \sum_{i=1}^{K} frecventa(c_i)$$

`abatere(`p_i`, `c_i`)` reprezintă abatarea dintre cuvântul inițial și cuvântul corect, iar `frecventa(`c_i`)` este frecvența cuvântului c_i, conform dicționarului.

Totuși, unii utilizatori sunt neglijenți și uită să pună spații între unele sau chiar între toate cuvintele. Este chiar posibil ca ei să pună spații în plus, între caracterele cuvintelor. Corectarea independentă nu mai funcționează și algoritmul trebuie îmbunătățit.

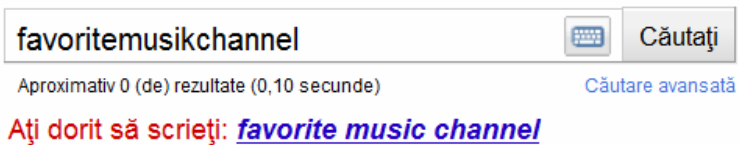

Figura 2. Motorul de căutare va sugera corecții plauzibile chiar și în cazul căutării alăturate.

Pentru a oferi rezultate relevante în astfel de situații, corectorul elimină toate spațiile din șirul original, păstrând ordinea celorlaltor caractere, după care introduce spații în anumite poziții, obținând un nou șir de cuvinte (posibil șirul de la care s-a plecat, dacă spațiile sunt introduse în pozițiile inițiale). Nu se pot introduce spații la începutul sau la sfârșitul șirului, și nu se va introduce mai mult de un spațiu în aceeași poziție. Pentru șirul astfel împărțit pe cuvinte se calculează abaterea totală și frecvența totală, conform descrierii de mai sus. Dintre toate posibilitățile de a insera spații, se va alege una după următoarele criterii:

- se va alege posibilitatea pentru care se obține o abatere totală minimă (1)
- dacă există mai multe posibilități, dintre cele cu abaterea minimă se va alege cea care conține un număr minim de cuvinte (2)
- dacă și în acest caz există mai multe posibilități, dintre variantele rămase la punctul (2) se alege cea cu frecvența totală maximă (3)

- dacă și la punctul (3) există mai multe posibilități, se va alege cea în care șirul obținut este minim lexicografic (cât mai mic alfabetic, spațiul fiind mai mic decât orice literă) (4)

Realizați un program care oferă o corecție pentru un șir dat, pe baza unui dicționar, conform regulilor de mai sus.

6. Structura datelor de intrare și de ieșire

Fișierul dict.txt va conține dicționarul, a cărui structură este descrisă la punctul 2. Acest fișier se va afla în același director în care se află și executabilul evaluat. Pentru a deschide dicționarul pentru citire, va trebui să se specifice calea lui relativă, și nu cea absolută. De exemplu, în C puteți folosi:

```
FILE *f = fopen("dict.txt", "r");
```

Șirul de caractere care va trebui corectat, conform indicațiilor de la punctul anterior, va fi citit de la tastatură (*standard input*). Acest șir va fi format numai din litere mici ale alfabetului englez și din spații. Șirul se va termina cu sfârșit de linie (\n) și nu va conține două spații alăturate.

Corecția sugerată pentru șirul citit va fi afișată pe ecran (*standard output*).

7. Exemple

standard input	**standard output**
error free text	error free text
this is dumy text	this is dummy text
supermarket	supermarket
sore served	so reserved
solvethis coo lhomewor	solve this cool homework
schaprobl mcanbe easilisolved	such a problem can be easily solved

Atât în exemplele de mai sus, cât și în testele ce vor fi folosite în corectarea temei, dicționarul considerat este cel de pe cs.curs.pub.ro.

8. Restricții și precizări

- Șirul care trebuie corectat este format din maxim 64 de caractere
- Timp maxim de execuție pe test:
  - **2.5 secunde** pentru implementările în C/C++
  - **3.5 secunde** pentru implementările în Java