Analiza Algoritmilor
Tema - Etapa 1

Grigore Lucian-Florin

Grupa 324CD
Facultatea de Automatica si Calculatoare
Universitatea Politehnica, Bucuresti

1 Descrierea problemei rezolvate

Problema abordata presupune gasirea celei mai lungi secvente comune pentru doua siruri de caractere. De asemenea, ne intereseaza complexitatea pentru fiecare solutie propusa daca am sti ca cea mai lunga secventa comuna are lungimea M.

2 Specificarea solutiilor alese

Solutiile alese pentru rezolvarea cerintei sunt urmatorii algoritmi

- **Rabin-Karp**
  Dorim sa gasim probabilitatea unei false potriviri in functie de dimensiunea spatiului cheilor, precum si folosirea mai multor tipuri de functii hash in implementare. In schimb, se pot elimina foarte usor secventele care nu se potrivesc cu ajutorul functiei hash. In cazul unui "false positive", dorim sa verificam potrivirea dintre fiecare litera a celor doua secvente, pentru a fi siguri de aceasta (parcugem cele doua secvente litera cu litera).

- **Knuth-Morris-Pratt**
  Dorim sa gasim doua probleme care poti fi rezolvate eficient cu acest algoritm, dar nu si cu Rabin-Karp. Aceasta solutie este mai eficienta in cazul cautarii unei singure secvente intr-un text, fata de prima solutie propusa. Acest algoritm presupune si construirea unei tabele speciale (denumita "failure function" sau "partial match table"), fiind de fapt un vector care sa ne ajute sa nu mai verificam un numar de caractere in cazul unei potriviri false.

3 Criteriile de evaluare pentru solutia propusa

Ne propunem sa evaluam cei doi algoritmi propusi dupa criteriile urmatoare:

- In cazul algoritmului **Rabin-Karp**, dorim sa gasim cazuri de functii hash care sa nu functioneze mereu, dar si alegerea unor alte functii hash care sa fie cat mai fail-proof.
- Cazurile cand algoritmul **Knuth-Morris-Pratt** functioneaza mai bine decat Rabin-Karp, si invers.
- Complexitatile fiecarui algoritm pentru un set de date de dimensiune cunoscuta *(cea mai lunga secventa fiind de o lungime M, cunoscuta)*.
- Dorim sa intocmim un **set de teste** care sa se asigure ca solutiile propuse functioneaza cat mai eficient din punctul de vedere al memoriei si timpului de executare. Testele propuse ar trebui sa acopere situatii precum: potrivirea multipla intre secvente de aceeasi lungime (preferabil lungime mare, pentru a vedea consumul resurselor), secvente care sa para la fel pentru functii hash slabe (de exemplu "aab" si "aba" ar avea aceeasi valoare hash pentru o functie care nu se foloseste de indexul literelor) sau secvente cu grupuri de litere repetitive care sa puna la incercare tabela caracteristica algoritmului KMP.

Referinte

1. Rabin-Karp Algorithm
   Link Wikipedia RB
   Link Brilliant RB

2. Knuth-Morris-Pratt Algorithm
   Link Wikipedia KMP
   Link Brilliant KMP

3. String Matching Algorithms Presentation
   Link

4. LCS problem
   Wikipedia Link