

Solutie Afaceri autor Airinei Adrian

Solutia 1: Impartim sirul in bucati de \sqrt{N} bucati si calculam in N^2 matricea $A[i][j]$ = numarul de pozitii in care difera subsecventele care incep la pozitiile i respectiv j de lungime \sqrt{N} . Un query il rezolvam in $O(\sqrt{N})$ iar un update in $O(n \cdot \sqrt{n})$. Observam ca atunci cand modificam un caracter se schimba in matrice maxim $\sqrt{N} \cdot N$ stari pe care le putem modifica in $O(1)$. Aceasta solutie obtine in functie de implementare in jur de 50 puncte.

Solutia 2, Mugurel Andreica :

Rafinand ideea impartirii sirului in bucati de \sqrt{N} vom numerota bucatile si vom calcula in N^2 matricea $A[i][j]$ = numarul de pozitii pe care difera subsecventele care incep la pozitia i de lungime \sqrt{N} si a j -a bucata de lungime \sqrt{N} . Un query il vom rezolva in $O(\sqrt{N})$ iar la un update observam ca se modifica cel mult N stari pe care le putem updata in $O(1)$.

Solutia 3, Mihai Patrascu:

Fie Si shiftarea cu i caractere spre dreapta a sirului, completata pana la lungimea $2 \cdot N - 1$ cu caracterele de la inceputul sirului. De exemplu, pentru sirul "alabalaportocala" (lungime 16):

| | |
|--------------------|---------------------------------|
| S0 (sirul initial) | alabalaportocalaalabalaportocal |
| S1 | alabalaportocalaalabalaportoca |
| S2 | alabalaportocalaalabalaportoc |
| S3 | alabalaportocalaalabalaporto |
| ... | |
| S15 | alabalaportocalaa |
| S16 | alabalaportocala |

Presupunand ca putem afla rapid raspunsul la subproblema

$SP(i, p1, p2)$ = distanta Hamming intre subsecventa $p1..p2$ a Si si subsecventa din sirul initial care se "suprapune" cu ea (vezi tabelul de mai sus pentru a intelege cum se suprapune o secventa dintr-o shiftare cu o secventa din sirul initial),

putem reduce la $SP(i, p1, p2)$ acele query-uri legate de doua subsecvente intre ale caror pozitii de inceput distanta este i . De exemplu, pentru o intrebare care ne cere distanta Hamming intre "abala" (incepe pe pozitia 3) si "porto" (incepe pe pozitia 8), vom apela $SP(5, 8, 12)$, deoarece $5 = 8 - 3$.

Pentru a rezolva subproblemele SP cu un anumit i vom folosi un arbore indexat binar in care al x -lea element e 1 daca si numai daca al x -lea caracter din Si e diferit de al x -lea caracter din S0 sau 0 altfel. Al i -lea AIB poate raspunde eficient la intrebarea "cate din primele x caractere ale lui Si difera de primele x caractere de pe pozitiile corespunzatoare din S0". In functie de caz, e nevoie de doua sau de trei apeluri la query-ul de AIB pentru a obtine raspunsul la un $SP(i, p1, p2)$. Complexitatea unei operatii Update (din problema originala) e $O(\log N)$.

Atunci cand avem de modificat un caracter, va fi nevoie de maxim doua operatii de update in fiecare dintre cele $N - 1$ AIB-uri. Complexitatea unei operatii Query (din problema originala) e $O(N \cdot \log N)$.