



PROBLEMA 2 – diez

Autor: Masterand Ștefan NEGRUȘ, Univ. Al. I. Cuza Iași

Solutia care implementeaza simularea operatiilor din enunt va trece aproximativ 4-5 teste. Aceasta solutie are complexitatea la inserare si interogare $O(\text{lungimea sirului})$.

O solutie optimizata presupune sa aflam raspunsul la interogari in mod offline.

Tinem toate operatiile in memorie si construim sirul final, dupa ce au loc toate inserarile. Pentru a face asta pornim cu un sir de caractere cu lungimea $N + \text{numarul de inserari}$ si initial doar cu caractere '#'. Parcurgem operatiile de la sfarsit catre inceput si, pentru fiecare operatie de inserare, cautam al **pos**-lea caracter '#' in acest sir si punem caracterul corespunzator inserarii pe acea pozitie. Dupa ce procesam inserarile, in sirul nostru de caractere vom avea '#' pe pozitiile unde ar trebui sa fie caracterele din sirul initial, in aceeasi ordine. Printr-o singura parcurgere putem completa aceste pozitii. Sirul obtinut este corect deoarece la fiecare inserare, in modul descris mai sus, caracterul este asezat pe pozitia pe care trebuie sa fie in sirul final. Pentru a gasi al **pos**-lea '#' in sir putem folosi un arbore indexat binar care sa ne spuna cate caractere '#' sunt in sir pana la o pozitie data. Peste acest arbore trebuie sa facem o cautare binara pentru a gasi pozitia dorita, aceasta fiind relativa.

Acum putem raspunde la interogari in aceeasi maniera. Parcurgem operatiile de la sfarsit catre inceput iar operatiile de inserare vor fi echivalente cu eliminarea caracterelor de pe pozitia respectiva. Pentru a elimina un caracter putem folosi aceeași optimizare cu arbore indexat binar si cautare binara pe care am folosit-o si la inserari. Pentru a raspunde la interogari putem folosi un arbore de intervale in care in fiecare nod memoram pentru intervalul corespunzator urmatoarele elemente: **hash1** - valoare hash doar a caracterelor diferite de '#', **hash2** - valoare hash doar a caracterelor diferite de '#', dar diferita de **hash1** si **cnt** - numarul de caractere diferite de '#' din interval. Un aspect important este ca, pentru fiecare interogare, trebuie sa gasim pozitiile corespunzatoare in sirul nostru, acest lucru poate fi rezolvat folosind arborele si cautarea pe care le utilizam la eliminari.

Pentru aceasta solutie, pentru valorile **hash1** si **hash2** putem reprezenta secventele intr-o baza mai mare decat 26 si memoram rezultatele modulo doua valori mari. Un exemplu concret:

BASE 73

MOD1 1000000007

MOD2 1000000033

Pentru doua teste, daca se mentine o singura valoare hash cu **MOD** = 2^{64} vor exista coliziuni si astfel raspunsul final va fi gresit.

Pentru a calcula valoarea **hash1** intr-un nod vom folosi:

$(\text{valoarea hash1 din fiul stang} * \text{BASE}^{\text{cnt din fiul drept}} + \text{hash1 din fiul drept}) \% \text{MOD1}$.

Ramane doar sa afisam raspunsurile pentru interogari in ordinea initiala.

Complexitate: $O(N * \log^2(N))$