

## Problema blis (autor prof. Dan Pracsu)

Pentru rezolvarea primei cerințe, există două situații:

a)  $k \geq$  lungimea șirului de biți. În acest caz numărul zecimal maxim este dat de întreg șirul convertit în baza 10.

b)  $k <$  lungimea șirului. În acest caz, evident că numărul maxim va avea exact  $k$  biți. Se construiește mai întâi numărul zecimal  $x$  cu primii  $k$  biți din șir, apoi se parcurge restul de  $N - k$  biți eliminându-se din  $x$  cel mai din stânga bit (din cei  $k$ ) și adăugându-se la dreapta bitul curent. La fiecare pas se actualizează maximum.

Complexitatea este  $O(N)$ , unde  $N$  este lungimea șirului de biți.

În continuare sunt prezentate două soluții de rezolvare pentru cerința a doua.

### Soluția 1 (aproximativ 70 puncte, Complexitate $O(N \times N \times k)$ )

Vom construi o matrice  $A$ , în care  $A(i, j) = k$  înseamnă:  $k$  este cea mai mică valoare ce poate fi capătul din dreapta unui subșir strict crescător de lungime  $j$  care s-a construit cu primii  $i$  biți. Pentru fiecare  $i$  între 1 și  $N$ , se construiește numărul zecimal  $x$  cu ultimii  $j$  biți ( $i \geq j \geq i - k + 1$ ), apoi se caută în stânga poziției  $j$  posibile valori care actualizează valorile minime ale capetelor subșirurilor strict crescătoare de lungimi 1, 2, 3, ...

De remarcat faptul că în matricea  $A$  fiecare coloană va avea mereu elementele ordonate strict crescător, la fel cum și la soluția clasică de determinare a subșirului strict crescător de lungime maximă în complexitate  $O(N \log N)$  se construiește un vector care este mereu ordonat strict crescător.

### Soluția 2 (100 puncte, Complexitate $O(N \times k \times (\log N + k))$ )

Folosim metoda programării dinamice „înainte”. Vom utiliza un vector  $BST[i]$  cu semnificația „cea mai mică valoare în care se termină un subsir strict crescător de lungime  $i$ ”.

Inițializăm  $BST[0] = -\infty$  și  $BST[i] = +\infty$  pentru  $1 \leq i \leq N_{\text{biti}}$ .

Semnificația lui  $BST[i] = +\infty$  este aceea că nu s-a obținut un subșir strict crescător de lungime  $i$ .

Să presupunem că am ajuns la un bit  $P$  și cunoaștem șirul  $BST$  construit până la bitul  $P-1$ .

Construim toate numerele de cel mult  $k$  biți începând cu bitul  $P$ . Fie  $N_j$  numărul format cu  $j$  biți (cei de pe pozițiile  $P \dots P+j-1$ ). Fie  $L$  cea mai mare lungime unde avem  $BST[L] < N_j$ .

Dacă  $BST[L+1] > N_j$  atunci cu siguranța se poate realiza o îmbunătățire la lungimea  $L + 1$ . Totuși momentul potrivit pentru a face aceasta îmbunătățire este abia când ajungem să procesăm bitul de pe poziția  $P+j-1$  de aceea doar memorăm că la poziția  $P+j-1$  valoarea  $N_j$  ar avea posibilitatea să facă un update pentru lungimea  $L+1$ . După ce se generează toate aceste update-uri pentru pozițiile  $P, P+1, \dots, P+k-1$ , se updatează  $BST$  cu toate update-urile primite de poziția  $P$ .

După ce s-au parcurs toți cei  $N$  biți soluția  $L_{\text{max}}$  poate fi depistată ușor ca fiind cea mai mare poziție unde  $BST[L] < \infty$ .

Analiza complexității: Pentru fiecare dintre cele  $N$  poziții (factor  $N$ ) avem de efectuat următoarele operații:

1. Generarea celor  $k$  numere (factor  $k$ ) ; pentru fiecare este necesară determinarea poziției  $L(\text{factor } \log N \text{ prin cautare binară})$ . Deci avem  $O(N \times K \times \log N)$

2. Aplicare update-urilor (cel mult  $K \times (K-1)/2$ ) deci avem  $O(N \times k \times k)$

Cumulat avem  $O(N \times k \times (\log N + k))$