Problema blis (autor prof. Dan Pracsiu)

Pentru rezolvarea primei cerinţe, există două situaţii:
a) k >= lungimea şirului de biţi. În acest caz numărul zecimal maxim este dat de întreg şirul convertit în baza 10.
b) k < lungimea şirului. În acest caz, evident că numărul maxim va avea exact k biţi. Se construieşte mai întâi numărul zecimal x cu primii k biţi din şir, apoi se parcurge restul de N – k biţi eliminându-se din x cel mai din stânga bit (din cei k) şi adăugându-se la dreapta bitul curent. La fiecare pas se actualizează maximul.
Complexitatea este O(N), unde N este lungimea şirului de biţi.

În continuare sunt prezentate două soluţii de rezolvare pentru cerinţa a doua.

Soluţia 1 (aproximativ 70 puncte, Complexitate O(N x N x 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 subşir strict crescător de lungime j care s-a construit cu primii i biţi. Pentru fiecare i între 1 şi N, se construieşte numărul zecimal x cu ultimii j biţi (i >= j >= i – k + 1), apoi se caută în stânga poziţiei j posibile valori care actualizează valorile minime ale capetelor subşirurilor 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 şi la soluţia clasică de determinare a subşirului strict crescător de lungime maximă în complexitate O(N log N) se construieşte un vector care este mereu ordonat strict crescător.

Soluţia 2 (100 puncte, Complexitate O(N x k x (log N + k))

Folosim metoda programării dinamice „înainte”. Vom utiliza un vector BST [i] cu semnificaţia „cea mai mică valoare în care se termină un subsir strict crescător de lungime i”.
Iniţializăm BST [0] = - infinit si BST [i] = + infinit pentru $1 \leq i \leq$ Nr_biti.
Semnificaţia lui BST [i]=+infinit este aceea că nu s-a obţinut un subşir strict crescător de lungime i.
Să presupunem că am ajuns la un bit P si cunoaştem sirul BST construit pâna la bitul P-1.
Construim toate numerele de cel mult k biti începând cu bitul P. Fie N_j numărul format cu j biţi (cei de pe poziţiile P...P+j-1). Fie L cea mai mare lungime unde avem BST[L] < N_j.
Dacă BST[L+1] > N_j atunci cu siguranţa se poate realiza o îmbunătătile la lungimea L + 1. Totuşi momentul potrivit pentru a face aceasta îmbunătăţire este abia când ajungem să procesăm bitul de pe poziţia P+j-1 de aceea doar memorăm că la poziţia P+j-1 valoarea N_j ar avea posibilitatea să facă un update pentru lungimea L+1. După ce se genereaza toate aceste update-uri pentru poziţiile P, P+1,...,P+k-1, se updateaza BST cu toate update-urile primite de poziţia P .
După ce s-au parcurs toţi cei N biţi soluţia Lmax poate fi depistată usor ca fiind cea mai mare poziţie unde BST[L]<infinit.

Analiza complexităţii: Pentru fiecare dintre cele N poziţii (factor N) avem de efectuat următoarele operaţii:

1. Generarea celor k numere (factor k) ; pentru fiecare este necesară determinarea poziţiei L(factor logN prin cautare binara). Deci avem O (N x K x logN)
2. Aplicare update-urilor (cel mult K*(K-1)/2) deci avem O(N x k x k)

Cumulat avem O(N x k x (log N +k))