Cifru – descriere solutie

prof. dr. Doru Popescu Anastasiu – C.N. ”Radu Greceanu”, Slatina
stud. Andrei Pârvu – Universitatea Politehnica București

Este clar ca o cifră din cifru nu poate apărea de mai multe ori decât numărul minim de apariții ale acesteia într-unul din numerele din fișierul de intrare. Astfel, număr în parte se determină frecventele de apariție ale cifrelor acestui numar folosind o matrice F[i,j]= frecventa cifrei j pentru numarul asociat liniei i, i=1, 2, ..., n.

Apoi pentru a determina cifrele folosite la cifru calculam: M[j] = minim{F[i,j] | i=1,2, ..., n}, j=1, 2, ..., 9, la cifru putând folosi cifra j de M[j] ori.

Soluția 1 – 10 puncte

Se folosește o tehnica de backtracking asemănatoare cu generarea permutarilor, asigurându-ne că se respectă restricțiile din M.

Soluția 2 – 30 puncte

Fie x[1], ..., x[p] componentele strict pozitive din M.

Numărul de numere cu 1, 2, ..., m cifre este egal cu o sumă de aranjamente cu repetiție, M = M[0] + ...+ M[9] = x[1] + ... + x[p].

Dacă notăm cu A(M, k, x[1], ..., x[p]) numărul de aranjamente ale unei multi-mulțimi cu M elemente luate câte k în care primul element apare de maxim x[1] ori, al doilea de maxim x[2] ori, ..., al p-lea element de x[p] ori, numarul cautat este:

Nr = A(M, M, x[1], ..., x[p]) + A(M, M-1, x[1], ..., x[p]) + ... + A(M, 1, x[1], ..., x[p]). Exista mai multe modalități pentru a calcula A(M, k, x[1], ..., x[p]), una dintre ele este urmatoarea:

A(M, k, x[1], ..., x[p]) = $k! \sum_{(a[1], a[2], ..., a[p])} \frac{1}{a[1]!\, a[2]! \cdots a[p]!}$, unde (a[1], a[2], ..., a[2]) este o solutie in multimea numerelor naturale a acuatiei $a_1 + a_2 + ... + a_p = k$, cu conditiile $a_i \leq x[i]$, i=1,2, ..., p. Implementarea acestei soluții se face tot printr-o metodă de tip backtracking.

Soluția 3 – 100 de puncte

Cifrul de lungime maximă va avea lungimea M, și rezultatul pentru el va fi M! / (M[1]! * M[2]! * ... * M[9]). Pentru o lungime N mai mică decât M, N = M - k va trebui să scădem k unități din factorialele de la numitor, la numărător având, bineînțeles, (M - k)!

Astfel, putem rescrie formula pentru N ca fiind (N! / (M[1]! * M[2]! * ... * M[9]!)) * (pos1 + pos2 + ... +post), unde posi = un produs format din k termeni ai factorialelor de la numitor pe care doresc să îi scad.

Pot calcula acest lucru folosind programare dinamică: P[i][j] - suma tuturor produselor de j termeni din primele i cifre de la numitor. Aflându-mă la cifra i pot selecta câți termeni să introduc în produs (intre 0 si M[i]) și, astfel, presupunând că am ales p termeni la pasul curent:

P[i][j] += P[i - 1][j - p] * M[i] * (M[i] - 1) * (M[i] – p + 1), dacă p > 0 sau
P[i][j] += P[i – 1][j], dacă p = 0.

Rezultatul pentru lungime k va fi (M - k)! / (M[1]! * M[2]! * ... * M[9]!) * P[9][k], iar rezultatul final va fi suma tuturor lungimilor k = 0, 1, …, M.

Soluția 3 – 80 de puncte, prof. Adrian Panaete

Notez sol[k][j] = numarul de coduri corecte de lungime k folosind cifre mai mici sau egale cu j.
atunci cu notatia M[i] = numarul maxim de i pe care il am cod vom avea urmatoarea dinamica inainte:

sol[k][j]*combinari(k+p,p) se aduna la orice sol[k+p][j+1] unde 0<=p<=M[j+1]. Explicatia e urmatoarea. Pentru orice solutie de tip sol[k][j] aleg sa mai pun p=0,1,2,...,M[j+1] cifre j+1. Se formeaza astfel o solutie de lungime k+p in care p pozitii sunt ocupate in combinari(k+p,p) moduri de cifrele j+1 iar restul de k cifre de solutiile de k cifre cel mult egale cu j.

Solutia finala va fi sol[1][9]+sol[2][9]+...+sol[M][9] unde M e numarul maxim de cifre din cod.