



Problema cifre – descrierea soluției

prof. Carmen Popescu, C. N. „Gh. Lazăr” Sibiu

Se citesc pe rând numerele, dar nu le vom memora, ci vom construi următoarele tablouri:

$nr[k]$ = numărul de cifre k care apar în toate numerele citite;

$t[k]$ = numărul de numere având k cifre

(Atenție să numărați și numerele egale cu 0!)

De exemplu pentru șirul 3120 400 1000 50 1 7 30 60 vom avea:

$nr[0]=9$ $nr[1]=3$ $nr[2]=1$ $nr[3]=2$ $nr[4]=nr[5]=nr[6]=nr[7]=1$

$t[1]=2$ $t[2]=3$ $t[3]=1$ $t[4]=2$

Pentru rezolvarea problemei se folosește o strategie Greedy, încercând să distribuim cifrele de 0, apoi cifrele 1 etc începând cu cifra unităților, apoi a zecilor etc. Pentru aceasta putem folosi două variante:

Varianta 1. (cifre1.cpp) – 80 puncte

Construim efectiv numerele folosind un vector a cu 100000 de componente. Considerăm cifrele unui număr numerotate de la dreapta la stânga începând cu 0. Notăm cu m numărul maxim de cifre din numerele date.

$x=t[m]$

Vom memora în $a[0], a[1], \dots, a[x]$ numerele de m cifre. Completăm cifra de ordin $m-1$ a acestor numere cu cele mai mari cifre disponibile (folosind vectorului nr)

Numerele de $m-1$ cifre sunt memorate în $a[x+1], a[x+2], \dots, a[x+nr[m-1]]$

Vom completa acum numerele cifra de ordin $m-2$ a numerelor $a[0], \dots, a[x+nr[m-1]]$

ș.a.m.d.

Completarea cifrelor de un anumit rang se va face de la dreapta la stânga, pentru a evita să punem un 0 pe cea mai semnificativă poziție a unui număr.

Varianta 2.

Vom construi pe rând fiecare număr începând cu cele având cele mai multe cifre. Pentru a face acest lucru, mai construim două tablouri suplimentare:

$a[i][j][k]$ = numărul de numere de j cifre care conțin pe poziția i cifra k

$b[i][j]$ = numărul de numere cu j cifre care nu au încă completată cifra de pe poziția i

Pentru exemplul de mai sus obținem:

$a[0][1][0]=2$

$a[0][2][0]=3$

$a[0][3][0]=1$

$a[0][4][0]=2$

adică am poziționat 8 cifre de 0 pe poziția unităților

$a[1][2][1]=3$

$a[1][3][0]=1$

$a[1][4][2]=1$

$a[1][4][3]=2$

adică am distribuit pe poziția zecilor o cifra de zero la numerele de 3 cifre, 3 cifre de 1 la numerele de două cifre și câte o cifră de 3 și de 4 la numerele de 4 cifre. etc.

După construirea acestui tablou, construirea numerelor se face plecând de la cea mai semnificativă cifră a celui mai mare număr.

Suma de la prima cerință se poate face fără a construi numerele, adunând valorile:

$a[i][j][k] * (10^i)^k$

pentru $i=0, 1, \dots$, numărul maxim de cifre-1

$j=1, 2, \dots$, numărul maxim de cifre

$k=1, \dots, 9$



Soluție – prof. Claudiu Gorea

Pentru toate cele N numere citite vom determina:

- $F[X]$ – numărul cifrelor X , folosite în numerele inițiale
- $C[1]$ – numărul de cifre de la unități
- $C[2]$ – numărul de cifre de la zeci

...

- $C[k_{\max}]$ – numărul de cifre situate pe poziția k_{\max}

Pornesc de la cifra 0 către cifra 9, punând cifrele în fața numărului, iar când adaug 0,

Verific și că la următoarea unitate (suntem la zeci, atunci la sute în pasul viitor etc)

mai punem ceva în fața lui (excepție făcând poziția unităților).

Dacă nu pot pune 0, plasez cea mai mică cifră disponibilă.

Soluție 3 – prof. Emanuela Cerchez

Reprezentarea informațiilor

```
#define NMAX 100001 //numarul maxim de numere
#define LGMAX 20 //lungimea maxima a numerelor
int nr[LGMAX]; //nr[i]=numarul de numere cu i cifre
int nrc[10];
//nrc[i]=numarul de cifre i folosite in scrierea celor n numere
int v[NMAX][LGMAX];
//pe linia i in matricea v retin cifrele numarului i
int lg[NMAX];
//lg[i]=lungimea celui de al i-lea numar construit.
```

În v numerele vor fi memorate în ordinea crescătoare a numărului de cifre.

Pasul 1

Citesc numerele succesiv și contorizez în vectorul nrc numărul de cifre folosite, respectiv în vectorul nr numărul de numere pentru fiecare lungime posibilă.

Pasul 2.

Construiesc un vector auxiliar s

```
int s[LGMAX];
s[1]=1; for (i=2; i<LGMAX; i++) s[i]=s[i-1]+nr[i-1];
Numerele de i cifre sunt plasate în v de la s[i] la s[i]+nr[i]-1
```

Pasul 3

Plasez zerourile necesare în numere de o cifră (acestea sunt singurele numere care pot să înceapă cu cifra 0):

```
for (i=1; i<=nr[1] && nrc[0]; i++) {nrc[0]--; lg[i]=1;}
```

Plasez celelalte zerouri în mod echilibrat, până le epuizez

```
for (i=1; i<LGMAX-1; i++) //plasez zerouri pe pozitia i
{if (!nrc[0]) break;
 for (j=s[i+1]; j<=n && nrc[0]; j++) lg[j]++, nrc[0]--;
}
```

Distribui apoi cifrele nenule descrescător

```
c=9; for (i=LGMAX-1; i>=0; i--)
//plasez cifra c pe pozitia i in numerele pentru care exista cifra i disponibila
for (j=s[i]; j<=n; j++)
{while (!nrc[c]) c--;
 if (lg[j]<i) {nrc[c]--; v[j][i]=c;}
```

La final construiesc numerele și calculez suma

```
for (i=1; i<=n; i++) //construiesc numarul v[i]
{lg[i]=0; for (j=LGMAX-1; j>0; j--) lg[i]=lg[i]*10+v[i][j];
 smax+=lg[i]; }
```