

Algoritmă backtracking

Se dă $n \in \mathbb{N}^*$. Să se determine toate numerele cu cifre distincte, a căror sumă este n . Soluțiile se vor afișa lexicografic.

Ex: $n=3$

102

12

120

se obține din prelungirea anterioară

201

21

210

3

30

Analiza

- 1) Org. şirului de soluții și domenii
în vectorul backtracking-ului punem direct
cifrele numerelor generate.

$$\begin{aligned}\text{Domenii} \quad x_1 &\in 1 \dots \min(g, n) \\ x_2 &\in 0 \dots \min(g, n - x_1) \\ x_3 &\in 0 \dots \min(g, n - (x_1 + x_2))\end{aligned}$$

general:

$$x_k \in (k=1? 1:0) \dots \min(g, n - \text{sum})$$

unde $\text{sum} = \text{param. suplimentar la funcția de back}$,
ce pe fiecare nivel reține suma termenilor anteriori:
 $x_1 + \dots + x_{k-1}$

2) Valid : să nu se repete cifra curentă x_k
(p. eficiență implementării cu red. de marcare)

3) Soluție : $sum + x_k == n$

4) În prelungire !

```

void back(int k, int sum)
{
    for (x[k] = (k == 1); x[k] <= min(9, n - sum); x[k]++)
        if (!used[x[k]])
        {
            used[x[k]] = 1;
            if (sum + x[k] == n) afish(k);
            back(k + 1, sum + x[k]);
            used[x[k]] = 0;
        }
}

```