Pašto ženklai

Sprendimą aiškinsime nagrinėdami sąlygoje pateiktą pavyzdį. Duotos vertės 1, 4, 12, 21, galima naudoti ne daugiau kaip 5 ženklus. Sudarome lentelę, kurioje tilptų visos galimos vertės (nuo 1 iki *maxint*). Joje žymėsimės kiek reikia panaudoti pašto ženklų, norint sudaryti atitinkamą vertę. Kol nesudarėme nei vieno rinkinio, lentelę užpildykime kokiu nors dideliu skaičiumi, pvz. 31. (tai yra didžiausias galimas pašto ženklų skaičius rinkinyje plius vienetas).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
0	31	31	31	31	31	31	31	31	31	31	31	31	31	...

Į lentelę įterpėme papildomą vertę, lygią 0. Jai sudaryti reikia 0 pašto ženklų. Nuo tos vertės ir pradėsime. Prie nagrinėjamos vertės eilės tvarka pridėkime visas turimas vertes ir pažymėkime lentelėje, kad joms gauti prireikė vieno pašto ženklo. Gausime:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
0	1	31	31	1	31	31	31	31	31	31	31	1	31	...

Imame kitą vertę (vieną) ir vėl pridedame visas galimas vertes. Prie naujai gautų verčių pasižymime, kad joms gauti reikia 2 ženklų.

Paėmę tolesnę vertę (du) ir pridėję turimų ženklų vertes, gausime tokią lentelę:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
0	1	2	3	1	2	3	31	31	31	31	31	1	2	...

Taip pildome lentelę, kol nagrinėjamai vertei gauti prireikia daugiau nei *m* ženklų. Įrašant ženklų skaičių į lentelę, būtina tikrinti, ar anksčiau gautas ženklų kiekis nebuvo mažesnis už dabartinį. Pavyzdžiui, pildydami paskutiniąją lentelę toliau, ir prie 3 pridėję 1, gautume, kad ketvertui sudaryti reikia jau 4 ženklų, nes trejetui gauti reikia 3 ženklų. Tačiau ketvertą jau buvome sudarę su mažesniu ženklų kiekiu – vienu ženklu. Todėl iš dviejų rinkinių renkamės mažesnįjį: prie ketverto paliksime vienetuką.

```
const MAX_N = 30;  { maksimalus skirtingų pašto ženklų verčių skaičius }
      MAX_M = 30;  { maksimalus ženklų kiekis rinkinyje }
type vertės = array [1..MAX_N] of integer;
     ženklai = array [0..maxint] of byte;
                                { jei pasirinktume integer tipą, pritrūktų atminties }

function didžiausia (n, m: integer; vertė: vertės)
                                                        : integer;
  var ž: ženklai;
      i, j: integer;
begin
  for i := 0 to maxint do
    ž[i] := MAX_M;     { dar nesudarėme nei vieno ženklų rinkinio }
  i := 0;
  ž[0] := 0;
  while ž[i] <= m do  { jei dar neviršijome ženklų limito rinkinyje }
    begin
      for j := 1 to n do
        if i + vertė[j] <= maxint
           then if ž[i + vertė[j]] > ž[i] + 1
                   then ž[i + vertė[j]] := ž[i] + 1;
      i := i + 1;
    end;
  didžiausia := i - 1;
  end; { didžiausia }
```