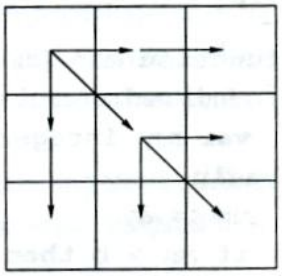
Lenta su skylėmis

Populiariausia šio uždavinio sprendimo idėja būna tokia: iš eilės imame visus lentos kvadratėlius ir bandome juos plėsti žemyn, į dešinę, stengdamiesi sudaryti kuo didesnį kvadratą be skylių. Taip patikrinę visus kvadratus, rasime didžiausiąjį.



19.2 pav.

Pasitelkus dinaminio programavimo metodą galima parašyti daug trumpesnį ir efektyvesnį algoritmą. Nagrinėkime sudarinėjamų kvadratų apatinius kairiuosius kampus. Sudarykime lentelę, turinčią tiek elementų, kiek lentoje yra langelių:

$$c[i, j], (1 \le i \le n, 1 \le j \le m),$$

čia $c[i, j]$ reikšmė būtų lygi didžiausio kvadrato be skylių, kurio apatinis dešinysis kampas yra langelyje (i, j), ilgiui. Jei langelyje (i, j) yra skylė, akivaizdu, kad $c[i, j] = 0$.

Jei šis langelis yra viršutinėje eilutėje ar pirmame stulpelyje tai $c[i, j] = 1$.

Kitais atvejais $c[i, j]$ reikšmė priklauso nuo to, kiek „erdvės“ yra nuo šio langelio į kairę, į viršų bei įstrižai. Tai galime sužinoti patikrinę kairėje, įstrižai ir viršuje esantiems langeliams atitinkančias lentelės c reikšmes. Iš trijų reikšmių išrinkę mažiausią ir ją padidinę vienetu, gausime ieškomąją $c[i, j]$ reikšmę.

Tai galima užrašyti formule:

$$c[i,j] = \begin{cases} 0, & \text{jei langelyje } [i,j] \text{ yra skylė;} \\ 1, & \text{jei } i = 1 \text{ arba } j = 1 \text{ ir langelyje skylės nėra;} \\ \min(c[i-1, j], c[i, j-1], c[i-1, j-1]) + 1, & \\ & \text{jei } i > 1,\ j > 1 \text{ ir langelyje skylės nėra.} \end{cases}$$

Vieną kartą peržiūrėję pradinę lentą, užpildome lentelę c. Sąlygoje pateiktam pavyzdžiui užpildyta lentelė atrodytų taip:

1	1	1	1	1	1	1	1	1
1	2	2	2	0	1	2	2	2
1	0	1	2	1	1	2	3	3
1	1	1	2	2	2	2	3	4
1	2	2	2	3	3	0	1	2
1	2	3	3	3	4	1	1	2

Norint sužinoti kvadrato, turinčio ilgiausią briauną, kraštinės ilgį, reikia peržiūrėti lentelę *c* ir išrinkti didžiausią reikšmę.

Beje, skaičiuojant *i*-osios eilutės elementų reikšmes naudojamasi jau apskaičiuotomis *i*-osios ir (*i*-1)-osios eilučių reikšmėmis. Taigi neverta įsiminti visos lentelės: užtenka dviejų eilučių – dabar pildomos ir prieš tai buvusios. O didžiausią reikšmę galime išrinkti iš karto pildydami lentelę.

```
const MAX = 100; { maksimalus lentos dydis }
type lenta = array [1..MAX, 1..MAX] of (kvadr, skylė);

function min (a, b, c: integer): integer;
{ randa mažiausią iš trijų skaičių }
  var mn: integer;
begin
  mn := a;
  if mn > b
     then mn := b;
  if mn > c
     then mn := c;
  min := mn
end;
```

```
function kvadratas (lent: lenta;
                    m, n: integer): integer;
  var did, i, j: integer;
      c1, c2: array [1..MAX] of integer; { viena lentelės c eilutė }
begin
  did := 0; { didžiausio kvadrato kraštinė }
  { užpildome pirmąją lentelės c eilutę }
  for j := 1 to n do
    if lent [1, j] = skylė
       then c1[j] := 0
       else begin
              c1[j] := 1;
              if c1[j] > did
                 then did := c1[j]
            end;
  { pildome kitas lentelės c eilutes }
  for i := 2 to m do
    begin
      for j := 1 to n do
        begin
          if lent[i, j] = skylė then c2[j] := 0
          else
            if j = 1 then c2[j] := 1
            else c2[j] := min (c2[j-1], c1[j-1], c1[j]) + 1;
          if did < c2[j]
             then did := c2[j]
        end;
      c1 := c2
    end;
  kvadratas := did
end;
```