

Žemėlapio spalvinimo uždavinio sprendimas kompiuteriu



Valentina Dagienė

dagiene@ktl.mii.lt

Bendrą sieną turinčių valstybių teritorijos politiniame žemėlapyje spalvinamos skirtinėmis spalvomis. Jau prieitame šimtmetyje buvo pastebėta, kad kiekvienam žemėlapiui nuspavinti pakanka keturių spalvų. Tačiau šią „keturių spalvų hipotezę“ pavyko įrodyti tik 1976 metais panaudojus kompiuterius. Straipsnyje aprašoma kompiutero programa, kuri, atitinkamai pateikus žemėlapio duomenis, kiekvienai šaliai parenka vieną iš keturių spalvų taip, kad bet kurios gretimos šalys būtų nuspavintos skirtingai.

Straipsnio autorė — MII vyresnioji mokslinė bendradarbė, docentė, daugiau kaip dviešimties knygų apie programavimą autorė.

Matematikai įrodė, kad bet kuriam politiniam žemėlapiui nuspavinti pakaks keturių spalvų¹. Tačiau norint iki galio atliskti darbą, reikia kiekvienai žemėlapio sričiai (valstybei) parinkti tinkamą spalvą. Tegul tai padaro kompiuteris. Sudarykime programą, kuri parinktų spalvas (raudoną, geltoną, mėlyną ir žalią) žemėlapio teritorijoms spalvinti.

Sudaromoji programa tiks bet kurios konfigūracijos žemėlapiui spalvinti: pasaulio, Europos, atskirų valstybių padalijimo į regionus ir pan. Dėl aiškumo rašydami programą turėkime omeny Lietuvos administracinį žemėlapį. Tai ypač svarbu galvojant apie pradinius duomenis, kurių šioje programoje nemažai.

Sprendžiant šį uždavinį, labai svarbu turėti tinkamai paruoštus pradinius duomenis (informaciją apie bendras rajonų sienas). Susitarkime, jog vieno rajono pradiniai duomenys yra vienoje duomenų eilutėje ir išdėstyti tokia tvarka:

- 1) rajono numeris;
- 2) rajono pavadinimas;

3) numerai tų rajonų, su kuriais rajonas turi bendras sienas; numerių sekos pabaigą žyni nulis.

Pradinius duomenis talpinsime vienmačiame masyve, kurio kiekvienas i -asis elementas yra kaimyninių rajonų aibė, t. y. aibė rajonų, turinčių bendras sienas su rajonu, kurio numeris i . Taigi informacija apie bendras rajonų sienas bus šitokiame masyve:

```
var sienos: array [Rajonai]
    of RajAibė;
```

čia tipai Rajonai ir RajAibė apibrėžti šitaip:

```
type Rajonai = 1..RSK; { rajonų skaičius }
RajAibė= set of Rajonai
```

Rajonų pavadinimus surašysime atskirai (jų reikės tik spausdinant rezultatus) eilučių masyve:

```
var RajVardai: array [Rajonai]
    of string[20]
```

¹ Žr. išsamiau apie šią problemą V. Stakėno straipsnyje „Keturios spalvos — ir pasaulis margas“, išspausdintą šiame žurnalo numeryje, p. 36–44.

Kad žinotume, kokiomis spalvomis nuspavinti rajonų teritorijas, sudarome masyvą, kurio kiekvienas elementas yra aibė indekso spalva spalvintų rajonų. Spalvos tipas apibrėžiamas taip:

```
type Spalva = (raudona, geltona,
                mėlyna, žalia);
```

pats masyvas aprašomas šitaip:

```
var spalvinama: array [Spalva]
                      of RajAibė;
```

```
procedure spalvinti (i: rajonai; var nuspavinta: boolean);
begin
```

imama pirmoji rinkinio spalva

repeat

if

ši spalva nebuvo panaudota
jokiam rajonui, turinčiam
sienas su *i*-uoju rajonu

then begin

šia spalva spalvinamas *i*-asis rajonas

```
if i = RSK { nuspavintas paskutinis rajonas }
```

```
then nuspavinta := true
```

```
else
```

```
begin
```

```
spalvinti (i + 1, nuspavinta)
```

```
if not nuspavinta then
```

ši spalva *i*-ajam rajonui netinka

```
end
```

```
end;
```

imama kita rinkinio spalva

until nuspavinta or

peržiūrėtas visas spalvų rinkinys

end;

Stačiakampiuose esantys veiksmai labai paprasti. Juos parašysime Paskalio kalba ir visą procedūrą iš karto įtrauksime į programą. Be to, sudarysime ir įtrauksime į programą procedūras pradiniams duomenims skaityti bei rezultatams rašyti. Jos ilgokos (ypač procedūra

Jau iš uždavinio formuluotės aišku, kad ji reikia spręsti bandymų metodu. Vadinasi, tekė sudaryti rekursinę procedūrą, kuri kiekvienam rajonui parinktų spalvą. Mus domina tik vienės sprendimo variantas — svarbu nurodytomis spalvomis nuspavinti Lietuvos Respublikos žemėlapį, o keliais būdais tai galima atlikti, neturi jokios reikšmės. Vadinasi, procedūra turi baigtis darbą, kai tik suranda bent vieną sprendinį. Pateikiame jos schemą:

```
program žemėlapis;
const RSK = 44; { rajonų skaičius; šie tiek rajonų Lietuvoje }
type Rajonai = 1..RSK;
RajAibė = set of Rajonai;
Spalva = (raudona, geltona, mėlyna, žalia);
```

skaityti), tačiau gana paprastos. Todėl detailiu neaiškinsime. Visose procedūrose masyvai sienos ir spalvinama yra globalieji kintamieji.

Pateikiame užbaigtą programą:

```

var prad, rez: text;
bvardas: string; { bylos vardas }
sienos: array [Rajonai] of RajAibė;
spalvinama: array [Spalva] of RajAibė;
rajvardai: array [Rajonai] of string[20];
sp: Spalva;
nuspavinta: boolean;

procedure spalvinti (i: Rajonai;
                     var nuspavinta: boolean);
var sp: Spalva;
begin
  sp := raudona;
  repeat
    if sienos[i] * spalvinama[sp] = []
    then begin { nėra spalva sp nuspavintų rajono t kaimyninių rajonų }
      { rajonas i spalvinamas sp spalva }
      spalvinama[sp] := spalvinama[sp] + [i];
      if i = RSK { nuspavintas paskutinis rajonas }
        then nuspavinta := true
      else begin
        spalvinti (i + 1, nuspavinta);
        if not nuspavinta
          then spalvinama[sp] :=
            spalvinama[sp] - [i]
        end
      end;
    { imama kita rinkinio spalva }
    if sp <> žalia
      then sp := succ(sp)
      else sp := raudona
    until nuspavinta or (sp = raudona)
  end; { spalvinti }

procedure skaityti;
{ skaitoma informacija apie rajonų sienas }
var numeris, k: Rajonai;
i: integer;
c: char;
begin
  for k := 1 to RSK do
    begin
      read(prad, numeris);
      rajvardai[numeris] := "";
      repeat { praleidžiami tarpai prieš pavadinimą }
        read(prad, c)
      until c <> ' ';
      while c <> ' ' do
        begin
          rajvardai[numeris] := rajvardai[numeris]+c;
          read(prad, c)
        end;
    { skaitoma informacija apie ribas tarp rajonų }
    sienos[numeris] := [];
    read(prad, i);
  end;

```

```

repeat
    sienos[numeris] := sienos[numeris] + [i];
    read(prad, i)
until i = 0
end
end;

procedure rašyti;
var i: Rajonai;
    sp: Spalva;
    j: integer;
begin
for i := 1 to RSK do
begin
    begin
        write(rez, rajvardai[i]);
        sp := raudona;
        while not (i in spalvinama[sp]) do
            sp := succ(sp);
        case sp of
            raudona: writeln(rez, 'raudona');
            melyna: writeln(rez, 'melyna');
            geltona: writeln(rez, 'geltona ');
            žalia: writeln(rez, 'žalia ')
        end
    end
end;
{ rašyti }

procedure bylos;
procedure pradbyla (var prad: text);
var bvardas: string;
begin
    write('Pradinių duomenų byla:');
    readln(bvardas);
    assign(prad, bvardas);
    reset(prad)
end;

procedure rezbyla (var rez: text);
var bvardas: string;
begin
    write('Rezultatų byla:');
    readln(bvardas);
    assign(rez, bvardas);
    rewrite (rez);
end;

begin { bylos }
    pradbyla(prad);
    rezbyla(rez)
end;
begin { žemėlapis }
    bylos;
    skaityti;
    nuspalvinta := false;
    for sp := raudona to žalia do

```

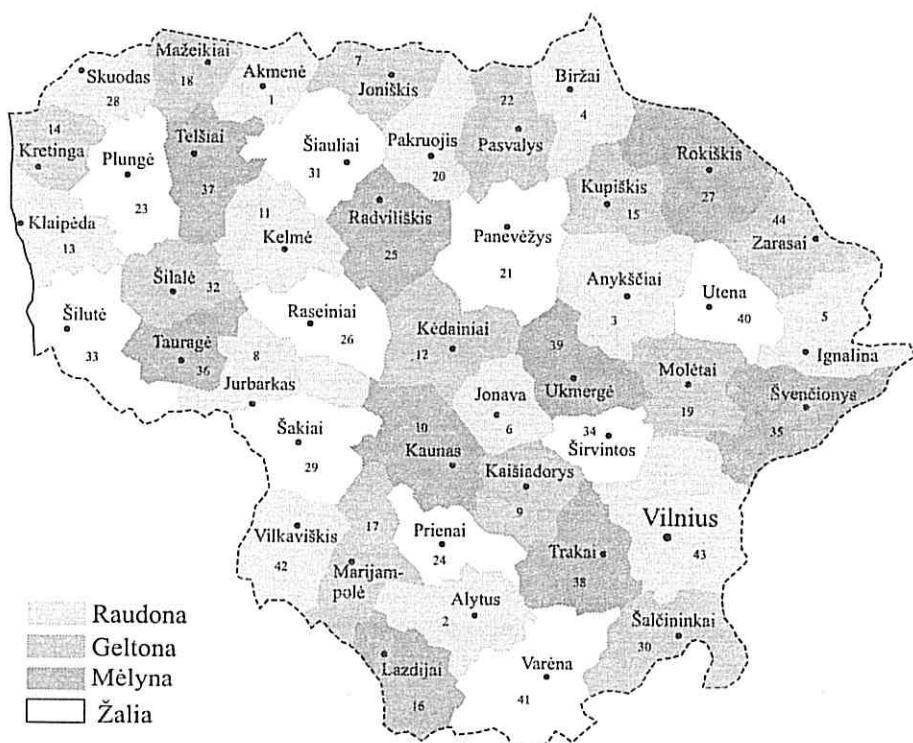
```

spalvinama[sp] := [];
spalvinti(1, nuspalvinta);
rašyti;
close(rez)
end.

```

Šiai programai pradiniai duomenys turėtų būti šitokie (rajonai sunumeruoti abécélės tvarka, jų išsidėstymas pateiktas paveiksle):

1	Akmenė	18	37	31	7	0
2	Alytus	17	24	38	41	16
3	Anykščiai	21	15	27	40	19
4	Biržai	22	21	15	27	0
5	Ignalina	40	44	35	0	
6	Jonava	12	10	39	34	9
7	Joniškis	1	31	20	0	
8	Jurbarkas	33	36	26	10	29
9	Kaišiadorys	10	6	34	38	24
10	Kaunas	26	8	29	12	6
11	Kelmė	37	32	36	31	25
12	Kėdainiai	26	25	21	39	6
13	Klaipėda	14	23	32	33	0
14	Kretinga	28	13	23	0	
15	Kupiškis	21	4	27	3	0
16	Lazdijai	17	2	41	0	
17	Marijampolė	29	42	10	24	2
18	Mažeikiai	28	23	37	1	0
19	Molėtai	3	39	40	34	43
20	Pakruojis	7	31	25	21	22
21	Panėvėžys	20	25	22	4	15
22	Pasvalys	20	21	4	0	
23	Plungė	28	14	13	32	37
24	Prienai	10	17	9	38	2
25	Radviliškis	31	11	26	20	12
26	Raseiniai	32	36	11	25	12
27	Rokiškis	4	15	3	40	44
28	Skuodas	14	23	18	0	
29	Šakiai	8	10	17	42	0
30	Šalčininkai	38	41	43	0	
31	Šiauliai	37	1	7	20	25
32	Šilalė	13	33	23	37	11
33	Šilutė	13	32	36	8	0
34	Širvintos	39	6	9	19	38
35	Švenčionys	40	19	5	43	0
36	Tauragė	33	32	11	26	8
37	Telšiai	23	18	1	31	11
38	Trakai	9	24	2	34	41
39	Ukmergė	21	12	3	19	34
40	Utena	3	27	19	35	5
41	Varėna	2	16	38	30	0
42	Vilkaviškis	29	17	0		
43	Vilnius	19	34	38	35	30
44	Zarasai	27	40	5	0	



Programa išspausdino šiuos rezultatus:

Akmenė raudona	Lazdijai mėlyna	Šiauliai žalia
Alytus raudona	Marijampolė geltona	Šilalė geltona
Anykščiai raudona	Mažeikių geltona	Šilutė žalia
Biržai raudona	Molėtai geltona	Širvintos žalia
Ignalina raudona	Pakruojis raudona	Švenčionys mėlyna
Jonava raudona	Panevėžys žalia	Tauragė mėlyna
Joniškis geltona	Pasvalys geltona	Telšiai mėlyna
Jurbarkas raudona	Plungė žalia	Trakai mėlyna
Kaišiadorys geltona	Prienai žalia	Ukmergė mėlyna
Kaunas mėlyna	Radviliškis mėlyna	Utena žalia
Kelmė raudona	Raseiniai žalia	Varėna žalia
Kėdainiai geltona	Rokiškis mėlyna	Vilkaviškis raudona
Klaipėda raudona	Skuodas raudona	Vilnius raudona
Kretinga geltona	Šakiai žalia	Zarasai geltona
Kupiškis geltona	Šalčininkai geltona	

Kaip matome iš gautų rezultatų, Lietuvos administraciniam žemėlapiui nuspalvinti prireiks visų keturių spalvų. Pirmiesiems pagal abėcėlę rajonams spalvinti daugiausiai vartojama raudona ir geltona spalvos, kadangi jos vardiniame tipe užrašytos pirmiau.

Be abejo, gražiau būtų, kad programa duomenis pateiktų vizualiai, t. y. tinkama spalva

nuspalvintų žemėlapį. Tačiau tam reikia ne mažai kruopštaus darbo. Siūloma skaitytojams pagalvoti, kaip tai būtų galima padaryti, projektą aprašyti ir pateikti sprendimą (galima programuoti ne tik Paskaliu, bet ir Delphi ar kitomis programavimo kalbomis, labiau tinkančiomis grafiškai vaizduoti duomenis).



Lietuvos žemėlapio taisyklingo spalvinimo būdas, kurį rado kompiuteris, nėra vienintelis. Sukeitę spalvas vietomis gautume $4! - 1 = 23$ spalvinimo variantus. Taigi galime sakyti, kad turime net 24 Lietuvos žemėlapio spalvinimo uždavinio sprendinius.

Pabandykite surasti dar vieną būdą. Jei rasite ir žemėlapį nuspalvinsite — turėsite spalvotą žurnalo egzempliorių, kokio galbūt niekas daugiau neturi!