KAUNO TECHNOLOGIJOS UNIVERSITETAS

Informatikos fakultetas

Sisteminės analizės katedra

Taikomosios diskrečiosios matematikos kursinis darbas

Atliko: IFI-2 gr. stud. M. Juzaitis

Projekto vadovas: dėst. M. Patašius

KAUNAS

2013

1. Užduotis

1. Šachmatai.

Yra daug legendų apie šachmatus. Viena legenda byloja, kad šachmatų išradėjas, supažindinęs karalių su šachmatais, paprašė kuklaus atlygio.
Karalius išradėjui pasakė:

„Paimk šachmatų lentą ir pastatyk ant jos žirgą. Ant to šachmatų langelio aš padėsiu $\boldsymbol{2^{k}}$ aukso monetų. Ant langelių, kuriuos žirgas gali pasiekti vienu ėjimu, aš padėsiu $\boldsymbol{2^{k-1}}$ aukso monetų. Ir apskritai, ant langelių į kuriuos žirgas gali patekti po ***p*** ($p \leq k$) ėjimų, aš padėsiu $\boldsymbol{2^{k-p}}$ aukso monetų. Tačiau, jei tu būsi per daug godus ir negalėsi pernešti visų lentoje sudėtų monetų, aš įsakysiu nukirsti tau galvą.“

Šachmatų išradėjas buvo labai protingas. Jis žinojo, kad gali panešti ne daugiau, kaip ***M*** monetų. Todėl išradėjas pastatė žirgą ant tokio langelio, kad gautų galimai daugiau monetų ir liktų gyvas. Jei tokio langelio duotam ***k*** nebūtų, tai išradėjas tyliai iškeliautų iš karalystės.

Sudarykite programą, kuri duotiems ***k*** ir ***M*** nurodytų visus langelius (jei tokių yra), ant kurių reikia statyti žirgą, kad padėtų monetų skaičius būtų galimai didžiausias, tačiau ne didesnis nei ***M***.

Įėjimo duomenys: $\boldsymbol{0 \leq k \leq 15,\ 1 \leq M \leq 10^{4}}$

Išėjimo duomenys: surinktų monetų skaičius, langeliai

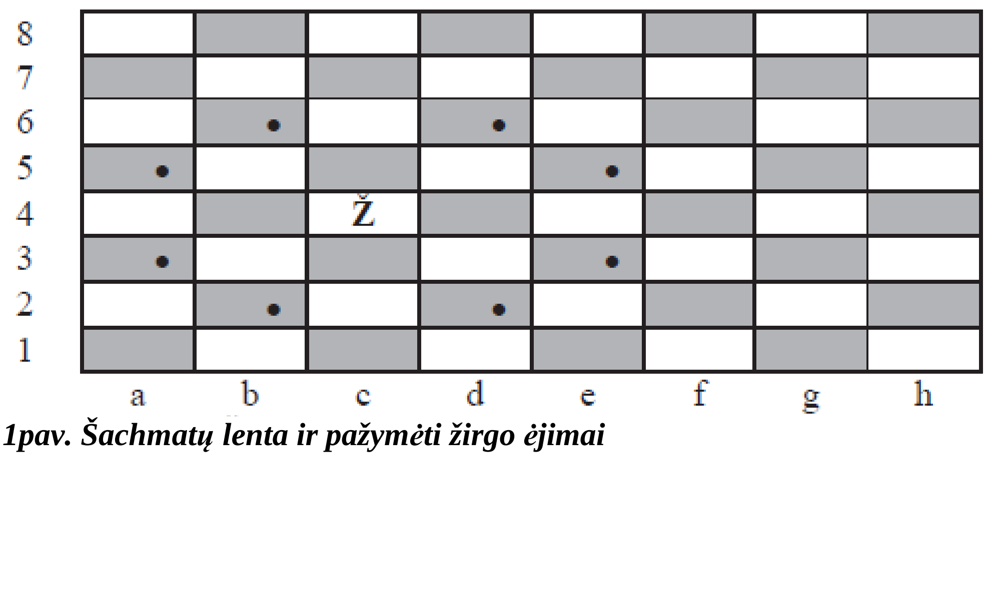
Pvz.

1) ***k=1***
M=5
surinktų monetų sk.=**5**
langeliai: **a2 a7 b1 b8 g1 g8 h2 h7**

2) ***k=2***
M=21
surinktų monetų sk.=**17**
langeliai: **a1 a8 h1 h8**

3) ***k=3***
M=92
surinktų monetų sk.=**91**
langeliai: **b3 b6 c2 c7 f2 f7 g3 g6**

4) **k=2**
M=4
Sprendinių nėra



1pav. Šachmatų lenta ir pažymėti žirgo ėjimai

2. Užduoties analizė

Jeigu šachmatų lentos langeliai yra grafo viršūnės (64 viršūnės) tai žirgo ėjimai yra to grafo briaunos. Atlikę visus galimus ėjimus gausime jungųjį grafą.

Duotojo uždavinio rezultatui gauti reikės atlikti veiksmus keliais etapais.

2.1. Pirmas etapas - Paieška platyn.

Atliksime trumpiausio kelio nuo viršūnės v_i iki likusių viršūnių radimą panaudojant paiešką platyn.

Paiešką pradedame iš viršūnės v_i. Nagrinėjame viršūnę v_i: ji tampa nenauja (aplankyta). toliau nagrinėjamos ir tampa nenaujomis visos viršūnės, gretimos viršūnei v_i, t.y. nagrinėjamos viršūnės, kurios nuo viršūnės v_i nutolę atstumu, lygiu 1. Po to nagrinėjamos ir tampa nenaujomis visos naujos viršūnės, gretimos prieš tai nagrinėtoms viršūnėms, t.y. nagrinėjamos visos naujos viršūnės, kurios nuo viršūnės v_i nutolę atstumu, lygiu 2. Apskritai, p-ajame žingsnyje nagrinėjamos ir tampa nenaujomis visos naujos viršūnės, gretimos (p-1)-ajame žingsnyje nagrinėtoms viršūnėms, t.y. viršūnės, kurios nuo viršūnės v_i nutolę atstumu, lygiu p. Paieška platyn baigiama, kai visos grafo viršūnės tampa nenaujomis, t.y. kai peržiūrimos visos viršūnės.
Paieškos platyn p-tojo žingsnio metu yra aplankomos viršūnės, kurių bendras skaičius n.

v_i viršūnė - pradinė paieškos viršūnė

n- viršūnių skaičius aplankytas p-tuoju žingsniu, paiešką pradedant nuo viršūnės v_i.

Kadangi žirgas per p ėjimų gali aplankyti visus langelius tai p yra baigtinis skaičius. Kadangi p reiškia atstumą nuo pirmos viršūnės iki toliausiai nutolusios, kuri buvo pasiekta trumpiausiu keliu, tai p(max) yra v viršūnės ekscentricitetas.

Suradę visų viršūnių ekscentricitetus galėsime rasti didžiausią, tokiu būdu nustatysime duotojo grafo skersmenį.

Kad rastume sprendimą, paiešką platyn turėsime atlikti iš kiekvienos grafo viršūnės. Turėsime nustatyti kelintu žingsniu pasiekta konkreti grafo viršūnė (suformuosime d masyvą kiekvienai viršūnei). Pagal d masyvą nustatysime kiek yra viršūnių aplankytų p-tuoju žingsniu ir kiekvienai suformuosime n masyvą.

n(v) masyvas: { $n_1 = 1; n_2 = \sum_{p=1} d_i; n_3 = \sum_{p=2} d_i; ...; n_{p+1} = \sum_{p=e_i} d_i$ }

n_1 – visada bus 1, nes visada paieška pradedama iš vieno langelio;

d_i – ilgis trumpiausio kelio nuo viršūnės *v* iki viršūnės i.

Viršūnes, kurių n masyvai vienodi, galime apjungti į vieną grupę, nes galutinis rezultatas iš šių viršūnių bus identiškas.

Viso susidaro dešimt viršūnių grupių:

Viršūnių grupė	Grupei priklausančios viršūnės	N(v) masyvas	Ekscentricitetas
g_1	a1; a8; h1; h8	$\{n_1=1; n_2=2; n_3=9; n_4=20; n_5=21; n_6=10; n_7=1\}$	$e_1=6$
g_2	b2; b7; g2; g7	$\{n_1=1; n_2=4; n_3=14; n_4=26; n_5=17; n_6=2\}$	$e_2=5$
g_3	c3; c6; f3; f6	$\{n_1=1; n_2=8; n_3=19; n_4=24; n_5=12\}$	$e_3=4$
g_4	d4; d5; e4; e5	$\{n_1=1; n_2=8; n_3=26; n_4=24; n_5=5\}$	$e_4=4$
g_5	a2; a7; b1; b8; g1; g8; h2; h7	$\{n_1=1; n_2=3; n_3=12; n_4=23; n_5=19; n_6=6\}$	$e_5=5$
g_6	b3; b6; c2; c7; f2; f7; g3; g6	$\{n_1=1; n_2=6; n_3=17; n_4=25; n_5=14; n_6=1\}$	$e_6=5$
g_7	c4; c5; d3; d6; e3; e6; f4; f5	$\{n_1=1; n_2=8; n_3=22; n_4=24; n_5=9\}$	$e_7=4$
g_8	a3; a6; c1; c8; f1; f8; f3; h6	$\{n_1=1; n_2=4; n_3=14; n_4=23; n_5=17; n_6=5\}$	$e_8=5$
g_9	b4; b5; d2; d7; e2; e7; g4; g5	$\{n_1=1; n_2=6; n_3=20; n_4=26; n_5=11\}$	$e_9=4$
g_{10}	a4; a5; d1; d8; e1; e8; f4; f5	$\{n_1=1; n_2=4; n_3=16; n_4=24; n_5=15; n_6=4\}$	$e_{10}=5$

2.2. Antras etapas – galutinio rezultato radimas.

Užduotyje duota, kad ant pradinės grafo viršūnės bus padėta 2^k monetų. Ant langelių, kuriuos galima pasiekti iš pirmojo langelio vienu žingsniu bus padėta 2^{k-1} monetų, dviem žingsniai 2^{k-2} monetų ir tt. Ant langelių, kuriuos galima pasiekti per p žingsnių 2^{k-p} monetų.

Tokiu būdų monetų sumą viršūnei v rasime pagal formulę

$$s(v) = n_1 \cdot 2^k + n_2 \cdot 2^{k-1} + n_3 \cdot 2^{k-2} + ... + n_{p+1} \cdot 2^{k-p}$$

$0 \le k \le 15$

Suformuojame masyvą $S(g_i)$, kuriame kiekvienai viršūnių grupei bus priskirtas monetų skaičius, kurį surinktume pradėję paiešką nuo tai grupei priklausančių langelių ir atlikę p žingsnių.

$S(g_i)$ masyvas $\{s(g_1); s(g_2); s(g_3); s(g_4); s(g_5); s(g_6); s(g_7); s(g_8); s(g_9); s(g_{10})\}$

Kad gautume galutinį rezultatą, mums reikės rasti didžiausią $S(g_i)$ masyvo reikšmę, kuri neviršytų M.

$1 \le M \le 10^4$

Jeigu visos masyvo reikšmės didesnės už M tai sprendinių nebus.

3. Programos tekstas

```
M=11;%Duotoji M reikšmė 1 <= k <= 10^4
k=1;%duotoji k reikšmė 0 <= k <= 15
n=64;%viršūnių skaičius
L=[11,18,12,19,17,13,20,18,9,14,21,19,10,15,22,20,11,16,23,21,12,24,22,13,23,14,3,19,26,4,20,27,25,5,21,28,26,17,1
,6,22,29,27,18,2,7,23,30,28,19,3,8,24,31,29,20,4,32,30,21,5,31,22,6,2,11,27,34,1,3,12,28,35,33,9,2,4,13,29,36,34,2
5,10,3,5,14,30,37,35,26,11,4,6,15,31,38,36,27,12,5,7,16,32,39,37,28,13,6,8,40,38,29,14,7,39,30,10,19,35,42,9,11,20
,36,43,41,17,10,12,21,37,44,42,33,18,11,13,22,38,45,43,34,19,12,14,23,39,46,44,35,20,13,15,24,40,47,45,36,21,14,16
,48,46,37,22,15,47,38,18,27,43,50,17,19,28,44,51,49,25,18,20,29,45,52,50,41,26,19,21,30,46,53,51,42,27,20,22,31,47
,54,52,43,28,21,23,32,48,55,53,44,29,22,24,56,54,45,30,23,55,46,26,35,51,58,25,27,36,52,59,57,33,26,28,37,53,60,58
,49,34,27,29,38,54,61,59,50,35,28,30,39,55,62,60,51,36,29,31,40,56,63,61,52,37,30,32,64,62,53,38,31,63,54,34,43,59
,33,35,44,60,41,34,36,45,61,57,42,35,37,46,62,58,43,36,38,47,63,59,44,37,39,48,64,60,45,38,40,61,46,39,62,42,51,41
,43,52,49,42,44,53,50,43,45,54,51,44,46,55,52,45,47,56,53,46,48,54,47]; %Suformuojam L masyvą
lst=[0,2,5,9,13,17,21,24,26,29,33,39,45,51,57,61,64,68,74,82,90,98,106,112,116,120,126,134,142,150,158,164,168,172
,178,186,194,202,210,216,220,224,230,238,246,254,262,268,272,275,279,285,291,297,303,307,310,312,315,319,323,327,3
31,334,336]; %Suformuojam lst masyvą
ATS=0;
BATS=[];
for i=1:n
  ats=0;
  Eile=[];
  Eile(1)=i;
  d=zeros(1,64); %Suformuojam d masyvas iš 64 nulių
```

```
  PirminesNuorodos=[];
  AntrinesNuorodos(1)=i;
  p=1;
  S=k;
  d(i)=1;
  tAts=2^k;
  while p<=S && length(AntrinesNuorodos)>0
            PirminesNuorodos=[];
            PPP=0;
            for j=1:length(AntrinesNuorodos)
            v=AntrinesNuorodos(j);
            L_nuoroda=lst(v+1);
            L_objektuSkaicius=lst(v+1)-lst(v);
            OBSK=L_objektuSkaicius;
            R=OBSK;
            OSB=0;
            while R>0
                  R=R-1;
                  SekancioNodoNr=L_nuoroda-R;
                  SekantisNodas=L(SekancioNodoNr);
                  if d(SekantisNodas)<1
                      PirminesNuorodos(end+1)=SekantisNodas;
                      Eile(end+1)=SekantisNodas;
                      d(SekantisNodas)=1;
                      tarpinisAts=2^(k-p);
                      OSB=OSB+tarpinisAts;
                  end;


             end;
             PPP=PPP+OSB;


             end;


       AntrinesNuorodos=PirminesNuorodos;


       ats=ats+PPP;
       p=p+1;


   end;
   PirminesNuorodos=[];
   AntrinesNuorodos=[];
   ats=ats+tAts;
   if ats<=M && ats>ATS
       ATS=ats;
       BATS=[];
       BATS(1)=i;


   elseif ats<=M && ats==ATS
       BATS(end+1)=i;


   end;
end;
% disp(BATS) %Išmeta viršūnių sąrašą
V1=['a1 a8 h1 h8']; %Viršūnių sąrašas
V2=['a2 a7 b1 b8 g1 g8 h2 h7'];
V3=['a3 a6 c1 c8 f1 f8 f3 h6'];
V4=['a4 a5 d1 d8 e1 e8 f4 f5'];
V10=['b2 b7 g2 g7'];
V11=['b3 b6 c2 c7 f2 f7 g3 g6'];
V12=['b4 b5 d2 d7 e2 e7 g4 g5'];
V19=['c3 c6 f3 f6'];
V20=['c4 c5 d3 d6 e3 e6 f4 f5'];
V28=['d4 d5 e4 e5'];
str = ['k=' num2str(k)];
disp(str)
strc = ['     M=' num2str(M)];
disp(strc)
if k==0
    str = ['     Surinktų monetų skaičius=' num2str(ATS)];
    disp(str)
       str = ['     Langeliai:' V1,' ',V2,' ',V3,' ',V4,' ',V10,' ',V12,' ',V19,' ',V20,' ',V28];
       disp(str)
elseif k==1
    switch BATS(1)
      case 1
       str = ['     Surinktų monetų skaičius=' num2str(ATS)];
       disp(str)
       str = ['     Langeliai:' V1];
       disp(str)
      case 2
       str = ['     Surinktų monetų skaičius=' num2str(ATS)];
       disp(str)
       str = ['     Langeliai:' V2];
       disp(str)
      case 3
       str = ['     Surinktų monetų skaičius=' num2str(ATS)];
```

```
        disp(str)
        str = ['     Langeliai:' V3,' ' ,V4,' ' ,V10];
        disp(str)
       case 11
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V11,' ',V12];
        disp(str)
       case 19
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V19,' ',V20,' ',V28];
        disp(str)
    end;
else
switch BATS(1)
    case 1
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V1];
        disp(str)
    case 2
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V2];
        disp(str)
    case 3
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V3];
        disp(str)
    case 4
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V4];
        disp(str)
    case 10
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V10];
        disp(str)
    case 11
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V11];
        disp(str)
    case 12
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V12];
        disp(str)
    case 19
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V19];
        disp(str)
    case 20
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V20];
        disp(str)
    case 28
        str = ['     Surinktų monetų skaičius=' num2str(ATS)];
        disp(str)
        str = ['     Langeliai:' V28];
        disp(str)
otherwise
    str = ['     Sprendinių nėra'];
    disp(str)
end
end
```

4. Testiniai pavyzdžiai

Buvo panaudoti du testiniai pavyzdžiai.

Pirmasis pavyzdys. Duotas k=1, M=11

Programos rezultatas:

k=1

M=11

Surinktų monetų skaičius=10

Langeliai:c3 c6 f3 f6 c4 c5 d3 d6 e3 e6 f4 f5 d4 d5 e4 e5

Pirmasis pavyzdys. Duotas k=5, M=440

Programos rezultatas:

k=5

M=440

Surinktų monetų skaičius=432

Langeliai:c3 c6 f3 f6

5. Išvados

Programa veikia teisingai.

6. Literatūros sąrašas

1. Matlab dokumentacija http://www.mathworks.se/help/documentation-center.html (žiūrėta 2013-12-17)
2. „Diskrečiųjų struktūrų“ modulis „Moodle“ aplinkoje http://vma.ktu.lt/course/view.php?id=764 (žiūrėta 2013-12-17)