KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

TAIKOMOSIOS INFORMATIKOS KATEDRA

DISKREČIOSIOS STRUKTŪROS (P170B008)

KURSINIS DARBAS

Užduoties nr. 11

Atliko:

IFF-1/6 gr. studentas
Lukas Kuzmickas

Priėmė:

Dėst. Martynas Patašius

KAUNAS

2022

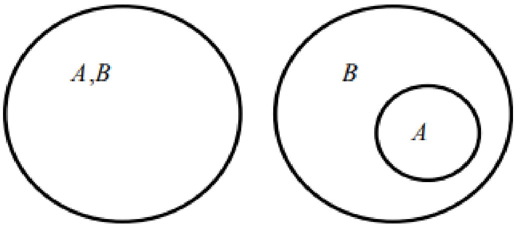
Turinys

1. Užduotis (Nd11)

Sugeneruoti visus duotos aibės poaibius. Realizuoti du metodus, iš kurių vienas generuotų Grėjaus kodus.

2. Užduoties analizė

Aibė A yra aibės B poaibis (žymime A ⊆ B), jei kiekvienas aibės A elementas yra ir aibės B elementas.



1 pav. Aibės ir poaibiai.

Grėjaus kodas – būdas sugeneruoti poaibius, kurie skiriasi tik vienu elementu (bitu). Hamingo atstumas tarp reikšmių yra lygus 1.

4 bit Gray Code	4 bit Binary Code
A B C D	B_4 B_3 B_2 B_1
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

2 pav. Dvejetainiai-Grėjaus kodai.

Uždavinys. Aibės A visų galimų poaibių aibė žymima P(A). Kitaip tariant . Pavyzdžiui, jei A = {1,2,3}, tai P(A) = {∅,{1},{2},{3},{1,2},{1,3},{2,3},{1,2,3}}.

Šios aibės dydis yra pastovus dydis: |A|=n , tai |P(A)| = 2^n.

Grėjaus kodas – reikia sugeneruoti visus poaibius, remiantis Grėjaus kodo taisyklėmis. Hamingo atstumas tarp reikšmių privalo būti lygus 1.

Pavyzdžiui, jei A = {1,2,3}, tai visi poaibiai sugeneruoti Grėjaus kodu atrodys taip:

P(GRAY) = {∅, {1},{1,2},{2},{2,3} ,{1,2,3},{1,3},{3}}.

Šiuos poaibius galime suprasti kaip bitų lentelę:

1	***2***	***3***
0	0	0
1	0	0
1	1	0
0	1	0
0	1	1
1	1	1
1	0	1
0	0	1

T.y. Grėjaus kodas, nes tarp dviejų reikšmių Hemingo atstumas yra lygus 1.

Metodo idėja.

Pirmojo metodo idėja - programa gaus aibę t.y. set masyvą. Įvedame dar papildomai masyvą b[0...n], kurį naudosime gretimiems poaibiams generuoti ir sąlyga t, prie kurios nustosime generuoti.

Antrojo metodo idėja – programa gaus aibę t.y. set masyvą. Įvedame dar papildomai masyvą b[0..n] ir kintamuosius *t, i*. Jei aibė A turi vieną elementą, tai Grėjaus kodai bus 0 ir 1. Grėjaus kodus aibei, turinčiai du elementus, generuosime remdamiesi Grėjaus kodais aibei, turinčiai vieną elementą. Turėdami Grėjaus kodus aibei, turinčiai (n-1) elementų, generuosime Grėjaus kodus aibei, turinčiai n elementų, tokiu būdu: 1. prie aibės, turinčios (n-1) elementą, Grėjaus kodų iš dešinės prirašome “0”, 2. po to prie aibės iš (n-1) elementų Grėjaus kodų, rašomų atvirkščia tvarka, dešinės pusės prirašome “1”. Tuo būdu gausime: jei n = 1, tai Grėjaus kodai yra 0, 1; jei n = 2, tai Grėjaus kodai yra 00, 10, 11, 01; jei n = 3, tai Grėjaus kodai yra 000, 100, 110, 010, 011, 111, 101, 001 ir t.t.

Funkcija Q(i), i ∈ N. Q(i) – tai toks didžiausias dvejeto laipsnis, kad $2^{Q(i)}$ yra i daliklis, t.y. *i* mod$2^{Q(i)}$ = 0.

Remdamiesi funkcija Q(i) sudarysime Grėjaus kodų generavimo algoritmą.

3. Programos algoritmo aprašymas

Pirmas metodas – poaibių generavimas leksikografine tvarka. Iš pradžių susigeneruojame pradinį kodą B masyve ir tada ieškome pirmojo iš dešinės elemento, kuris yra lygus nuliui. Generavimas baigiamas, kai masyvo b pirmojo elemento iš dešinės, indeksas bus lygus nuliui. Taip gauname visus aibės poaibius.

Antras metodas – poaibių generavimas Grėjaus kodais. Įėjimo duomenys, yra tokie pat, kaip ir algoritmo, generuojančio visus poaibius leksikografine didėjimo tvarka. Pradžioje turime Grėjaus kodo eilės numerius ir poaibių generavimo sąlygą t. Naudodami Q(i) funkcija einame per gretimus poaibius ir pagal tam tikras sąlygas invertuojame skiltis, generuodami poaibius Grėjaus kodo tvarka. Generavimas sustoja, kai pasiekiame 2^n elementų.

1 lentelė. Metodo aprašymo būdų palyginimas

Sprendimas	**Backtracking/Bitshifting**	**Leksikografine tvarka**	**Q(i) funkcijos**
1 metodas	Rekursija/sunkus	paprasta	-
2 metodas	Rekursija/sunkus	netinka	paprasta

[Gali būti verta pagrįsti, kodėl parinkti tokie įvertinimai.]

Kadangi šiuos atveju visi algoritmai, buvo duoti Moodle medžiagoje, juos galime lengvai implementuoti ir realizuoti Matlab pagalba. O kitos šio sprendimo funkcijos, reikalauja rekursijos arba bitshift operacijų.

[Panašiu palyginimu gali būti verta pagrįsti programavimo kalbos, algoritmo, jo dalies, duomenų struktūros, kieno nors dar pasirinkimą. Reikėtų, kad taip pasirenkama darbe būtų bent kas nors.]

4. Programos tekstas

allSubsets.m

```
clc; clear;
%Visos galimos aibės, algoritmas
set = [1,2,3];
n = length(set);
M(1:2^n,1:n)=0;
mi = 0;
b(1:n) = 0;
t = 1;
while(t)
    mi = mi + 1;
    fprintf('Poaibis: ');
    mj = 0;
    for i=1:n
       if (b(i) == 1)
            fprintf('%3d ',set(i));
            mj = mj + 1;
            M(mi,mj) = set(i);
       end
    end
    fprintf('\n');
    i = n;
    while(i >= 1 && b(i) == 1)
        b(i) = 0;
        i = i - 1;
    end
    if (i == 0)
        t = 0;
    else
        b(i) = 1;
    end

 end
% isveda M kaip matrica ir spalvu matrica
fig = uifigure;
uit = uitable(fig,'Data',M);
imagesc(M);
```

Poaibiu spausdinimui yra naudojama funkcija imagesc ir utable.

Imagesc – matricos vaizdavimas spalvomis.

Uitable – matricos vaizdavimas.

Funkcijos turinys nekeistas

graySubsets.m

```
clc; clear;
set = [1,2,3];
n = length(set);
M(1:2^n,1:n)=0;
mi = 0;
b(1:n) = 0;
i = 0;
t = 1;
while(t)
    mi = mi + 1;
    fprintf('Grėjaus poaibis: ');
    mj = 0;
    for k=1:n
       if (b(k) == 1)
            fprintf('%3d ',set(k));
            mj = mj + 1;
            M(mi,mj) = set(k);
       end
    end
    fprintf('\n');
    i = i + 1;
    p = 1;
    j = i;
    while(mod(j,2)==0)
        p = p + 1;
        j = fix(j/2);
    end
    if p<=n
        b(p) = 1 - b(p);
    else
        t = 0;
    end
end
%isveda M kaip matrica ir spalvu matrica
fig = uifigure;
uit = uitable(fig,'Data',M);
imagesc(M);
```

Poaibiu spausdinimui yra naudojama funkcija imagesc ir utable.

Imagesc – matricos vaizdavimas spalvomis.

Uitable – matricos vaizdavimas.

Funkcijos turinys nekeistas

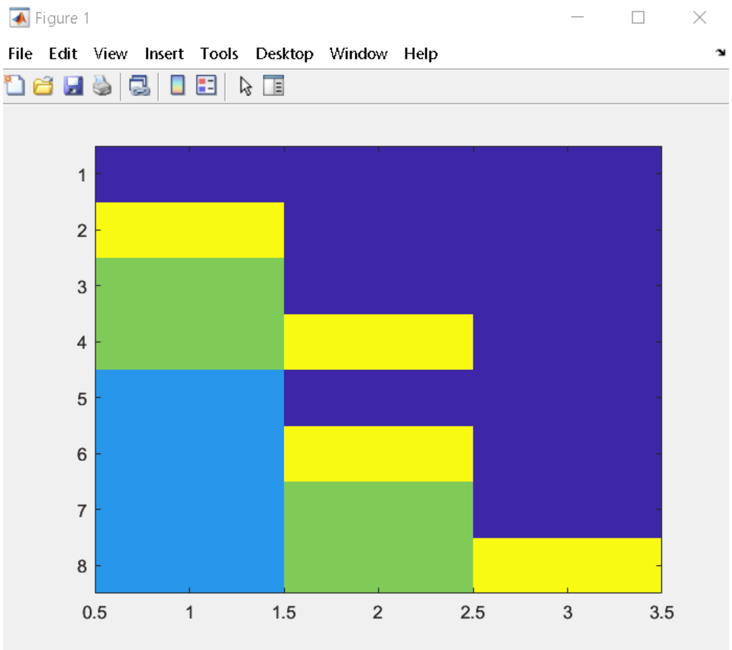
5. Testavimo pavyzdžiai

Buvo panaudoti trys testavimo pavyzdžiai (naudojame utable ir imagesc funkcijas).

Pirmas testas

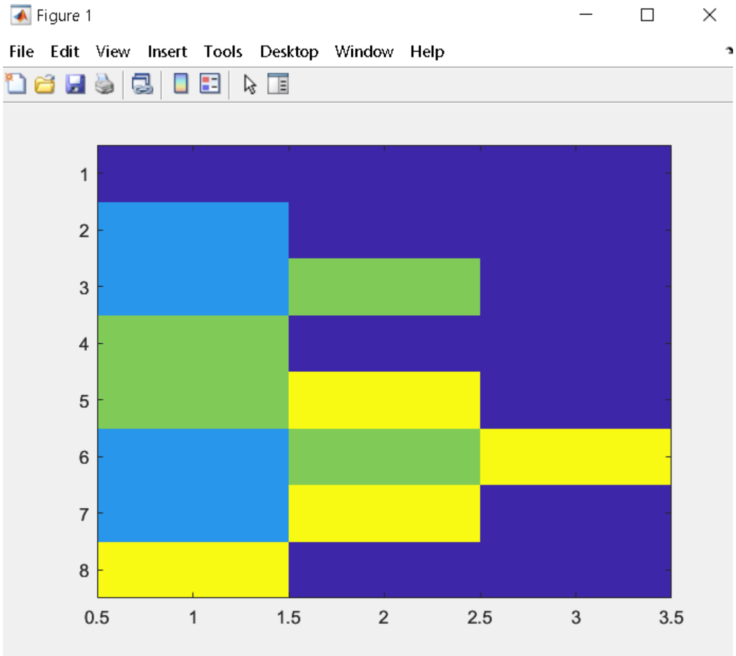
Naudojame tokia pradinę aibę:

```
B = [1,2,3];
```



	1	2	3
1	0	0	0
2	3	0	0
3	2	0	0
4	2	3	0
5	1	0	0
6	1	3	0
7	1	2	0
8	1	2	3

3 pav. 1 metodo rezultatai.



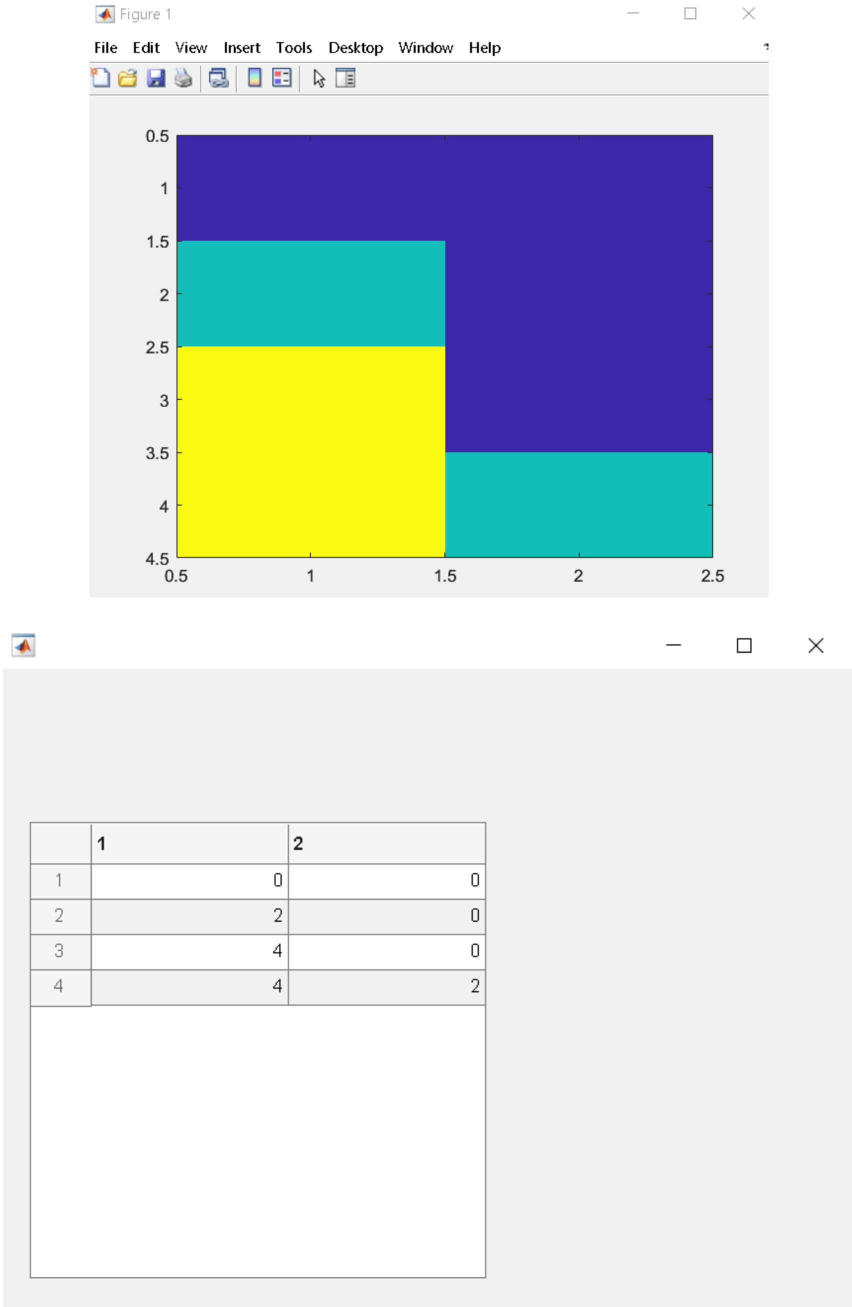
	1	2	3
1	0	0	0
2	1	0	0
3	1	2	0
4	2	0	0
5	2	3	0
6	1	2	3
7	1	3	0
8	3	0	0

4 pav. 2 metodo rezultatai.

Antras testas

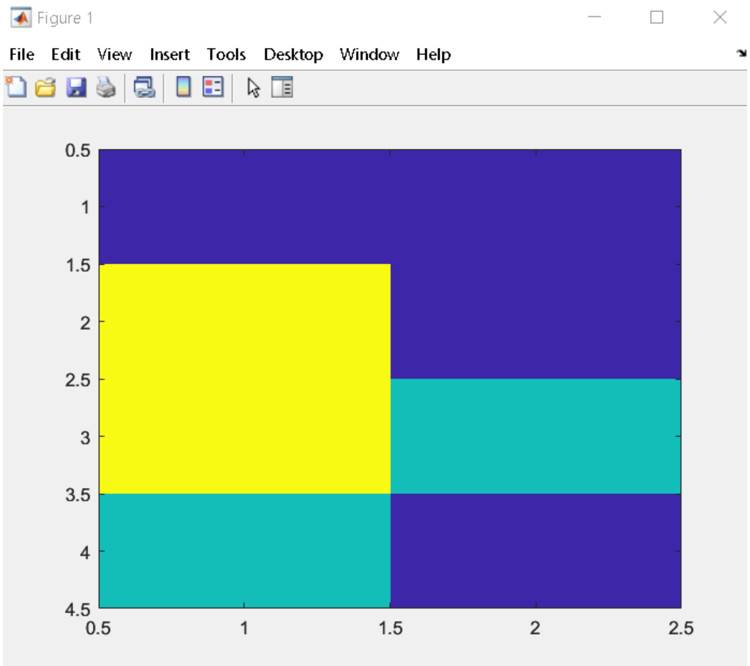
Naudojame tokia pradinę aibę:

```
B = [4,2];
```



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

5 pav. 1 metodo rezultatai.



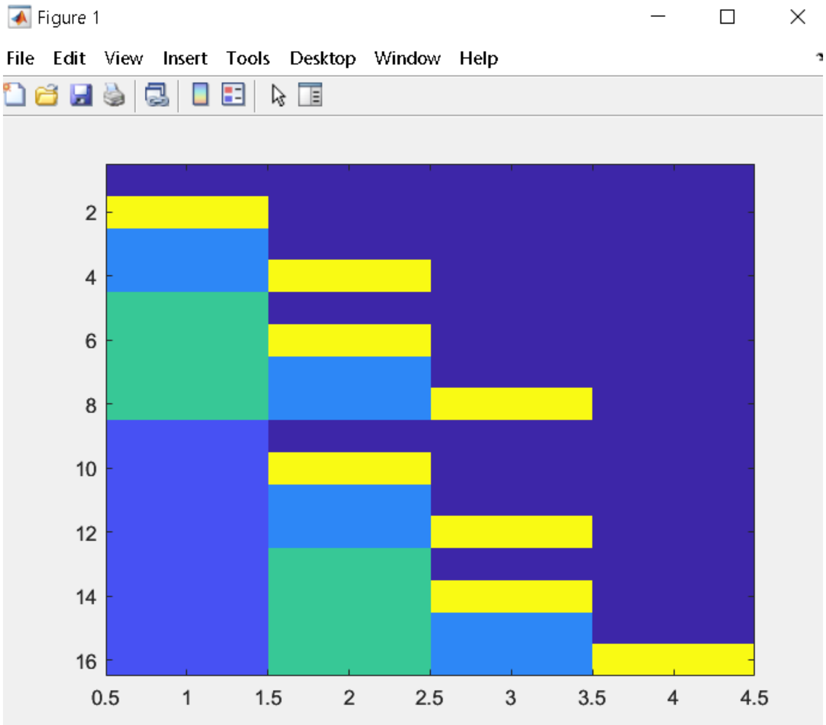
	1	2
1	0	0
2	4	0
3	4	2
4	2	0

6 pav. 2 metodo rezultatai.

Trečias testas

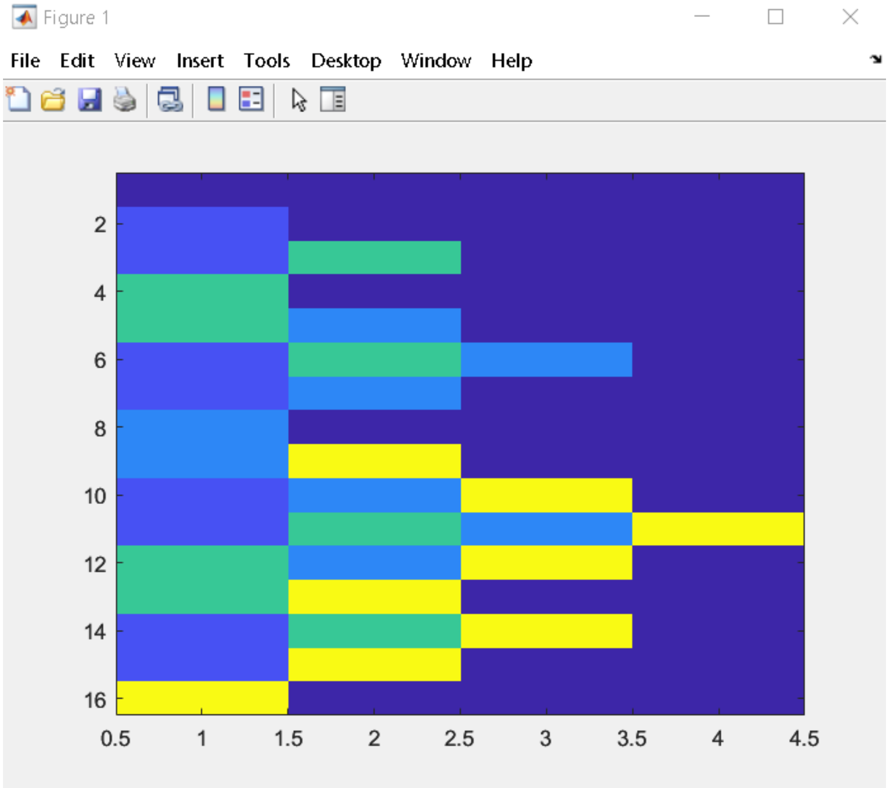
Naudojame tokia pradinę aibę:

```
B = [1,4,2,7];
```



	1	2	3	4
1	0	0	0	0
2	7	0	0	0
3	2	0	0	0
4	2	7	0	0
5	4	0	0	0
6	4	7	0	0
7	4	2	0	0
8	4	2	7	0
9	1	0	0	0
10	1	7	0	0
11	1	2	0	0
12	1	2	7	0
13	1	4	0	0
14	1	4	7	0
15	1	4	2	0
16	1	4	2	7

7 pav. 1 metodo rezultatai.



	1	2	3	4
1	0	0	0	0
2	1	0	0	0
3	1	4	0	0
4	4	0	0	0
5	4	2	0	0
6	1	4	2	0
7	1	2	0	0
8	2	0	0	0
9	2	7	0	0
10	1	2	7	0
11	1	4	2	7
12	4	2	7	0
13	4	7	0	0
14	1	4	7	0
15	1	7	0	0
16	7	0	0	0

8 pav. 2 metodo rezultatai.

6. Išvados

Programa veikia teisingai, išmokta realizuoti jungiojo grafo radimą naudojantis MATLAB bibliotekomis. Per šį kursinį darbą buvo susipažinta stipriai su MATLAB pradmenimis, pagilintos žinios apie aibes bei bendrąsias bibliotekų funkcijas.

Literatūros sąrašas

1. Matlab dokumentacija http://www.mathworks.se/help/index.html (žiūrėta 2022-11-08)
2. „Diskrečiųjų struktūrų“ modulis „Moodle“ aplinkoje https://moodle.ktu.edu/course/view.php?id=39 (žiūrėta 2022-11-12)