**Kauno technologijos universitetas**

Informatikos fakultetas

Taikomosios informatikos katedra

**Diskrečiosios struktūros (p170b008)**

**Kursinis darbas**

**Užduoties nr. A20**

Atliko:

IFF-0/3 gr. Studentė

Simona Ragauskaitė

Priėmė:

Doc. Martynas Patašius

KAUNAS

2021

Turinys

[1. Užduotis (A20) 3](#_Toc88581414)

[2. Užduoties analizė 3](#_Toc88581415)

[3. Programos algoritmo aprašymas 3](#_Toc88581416)

[4. Programos tekstas 3](#_Toc88581417)

[5. Testavimo pavyzdžiai 5](#_Toc88581418)

[Pirmas testas 5](#_Toc88581419)

[Antras testas 6](#_Toc88581420)

[Trečias testas 7](#_Toc88581421)

[6. Išvados 8](#_Toc88581422)

[7. Literatūros sąrašas 8](#_Toc88581423)

# Užduotis (A20)

Parduotuvėje yra n rūšių pyragaičių. Keliais skirtingais būdais galima nusipirkti k pyragaičių? Išvardinkite tuos būdus. Palyginkite sprendimą (trukmę ir rezultatus) su standartinės funkcijos rezultatu.

# Užduoties analizė

**Metodo idėja**. Tarkime, kad pyragaičių rūšis žymėsime pirmosiomis raidėmis. Jas apsirašysime į masyvą. Taip pat, nurodysime po kiek norime jų nusipirkti, kad galėtume apskaičiuoti ir surasti galimus pasirinkimo variantus. Norėdami surasti keli pasirinkimo variantai yra galime apskaičiuoti naudodami derinių formulę. O kokie galimi pasirinkimo deriniai yra apskaičiuosime rekursiniu būdu bei išspausdinant visus variantus. Taip pat, norėdami patikti sprendimo trukmę, palyginsime savo sukurtus metodus, panaudosime matematines formules su programinės įrangos „Matlab“ jau aprašytomis funkcijomis.

# Programos algoritmo aprašymas

Programa gaus „elements“ - pyragaičių rūšių masyvą raidžių (pvz.: elements = [“k”, “l”, “m”, “n”], rūšį žymėsiu viena raide). O n – apsiskaičiuosiu pagal kiek buvo duota rūšių.

Programa gaus „k“ - pyragaičių kiekį (pvz.: k = 2).

Programa, keliais skirtingais būdais galima išsirinkti, apskaičiuos pagal derinių formulę bei „Matlab“ funkcija, kad pamatytume metodo trukmę ir galėtume palyginti.

Programa išvardins galimus variantus parašytu rekursiniu metodu („Recursive.m“). Į šią funkciją atsisiunčiu tuščią eilutę („line“), rūšis („elements“), rūšių kiekį („n“), po kiek rūšių reikės suderinti („k“) ir įsivedu dar vieną elementą („l“) – apsaugo nuo variantų, kurie yra tie patys, bet skiriasi išdėstymo tvarka, jam priskiriu 0 reikšmę, kad išvengti pasikartojimų ir laikau – nusisiunčiu atsakymus („table“). Norėdama pereiti per visus elementus užsuku ciklą, ir pradedu tikrinti ar jau yra - buvo tokia reikšmė, jei ne, į tuščią eilutę talpinu pirmąją rūšį. Įdėjusi tikrinu ar jau nėra pasiektas reikiamas kiekis derinimų rūšių, jei ne – kartojame šią funkciją vėl, jei taip – išspausdiname tą derinį, bei jį susiunčiu į lentelę („Table.m“), kad jį patalpintų. Po šių sąlygų, iš eilutės variantų pašalinu paskutinį pridėtą elementą, kad pereitume visus variantus.

Programa išvardins galimus variantus su „Matlab“ funkcijomis aprašytą funkciją bei derinių kiekį.

Toliau yra išrašomi visi gauti rezultatai bei jų atlikimo trukmė.

# Programos tekstas

***Programa.m***

clc; close all; clear

%----------------Duomenys---------------

k = 1;

elements = ["k"];

%----------------Konstanta--------------

l = 1;

line = '';

n = length(elements);

chr = convertStringsToChars(elements);

%-----------------Programa--------------

%------Grafines sasajos deriniu lentele-

fig = uifigure(); % sukuriama erdve grafinei lentelej

uit = uitable(fig, 'Data', array2table(zeros(0, n+1))); % nubraižoma lentele

pause(1)

if n > 1

for i = 2:n+1

uit.ColumnName(i) = chr(i-1); % ? pirm? eilut? surašomos galimo reikšm?s

end

else

uit.ColumnName(2) = cellstr(chr(1));

end

uit.ColumnName(1) = {'Nr.'};

%---------------------------------------

disp('v Galimi variantai')

tic

Recursive(line,elements,n,k,l,uit);

toc

disp('^ Rekursinio metodo trukme')

disp('v Galimi variantai')

combos = matlabComb(elements,k);

combosConvert = cell2mat(combos);

disp(combosConvert);

tic

cc = matlabComb(elements,k);

toc

disp('^ Matlabo deriniu apskaiciavimas')

disp('v Pagal deriniu matematine formule apskaiciuotas deriniu kiekis')

count1 = factorial(n+k-1)/(factorial(n-1)\*factorial(k));

disp(count1)

tic

ccc = factorial(n)/(factorial(mod(n-k,1))\*factorial(k));

toc

disp('^ Trukme matematines deriniu formules')

disp('v Pagal Matlabo formule apskaiciuotas deriniu kiekis')

count = nchoosek(n+k-1,k);

disp(count)

tic

cc2 = nchoosek(n+k-1,k);

toc

disp('^ Trukme Matlabo formules');

***Recursive.m***

function re = Recursive(line,elements,n,k,l,table)

% suku cikla per visus elementus()

for i = l:n

% isidedu elementa

line = line + elements(i);

% tikrinu ar atitinka kiek galima pasirinkti kiekis

if strlength(line) == k

% isspausdinu i 'Command Window' galima pasirinkimo varianta

disp(line);

% papildau lentele vienu deriniu

table = Table(elements,n,table,line);

else

% kadangi neatitinka salygos kartoju rekursini metoda

Recursive(line,elements,n,k,i,table)

end

% išima iš derinio paskutin? pridet? element?, jog sekanti gal?tume patikrinti

line = extractBetween(line, 1, strlength(line)-1);

end

return

***Table.m***

function [table] = Table(elements, n, table, line)

lineArray = convertStringsToChars(line);

newRow = zeros(1,n); % susikuriu masyva su 0-ais - nauju elementu masyva

for i = 1:n % einame per elementus

for j=1:strlength(line) % einu per isrinktu variantus

if lineArray(j) == elements(i) % tikriname ar isrinktame variante yra tam tikras elementas

newRow(i) = newRow(i) + 1; % jei yra, irasome 1 toje vietoje

end

end

end

% is lenteles pasiimame visus duomenis

data = get(table,'Data');

if size(data) == [0, n+1] % kai nera nei vienos duomenu eilutes

newRow = cat(2, 1, newRow); % pridedam 1 pradžioje pirmos eilutes

table.Data = array2table(newRow); % prilyginame, irasydami newRow reiksmes

else

data(end+1,:)= {0}; % uzpildome pirmaja eilute 0

data(end,1) = num2cell(height(data)); % isidedame derinio numeri

data{end,2:n+1} = newRow(1:n); % perrasome informacija is newRow

set(table,'data', data);

end

return

***matlabComb.m***

function combs = matlabComb(values, k)

n = numel(values);% pasiimu elementu skaiciu

combs = bsxfun(@minus, nchoosek(1:n+k-1,k), 0:k-1); % atimame

combs = reshape(values(combs),[],k); % sudeda elementus po k ? eilutes

end

# Testavimo pavyzdžiai

Buvo panaudoti trys testavimo pavyzdžiai:

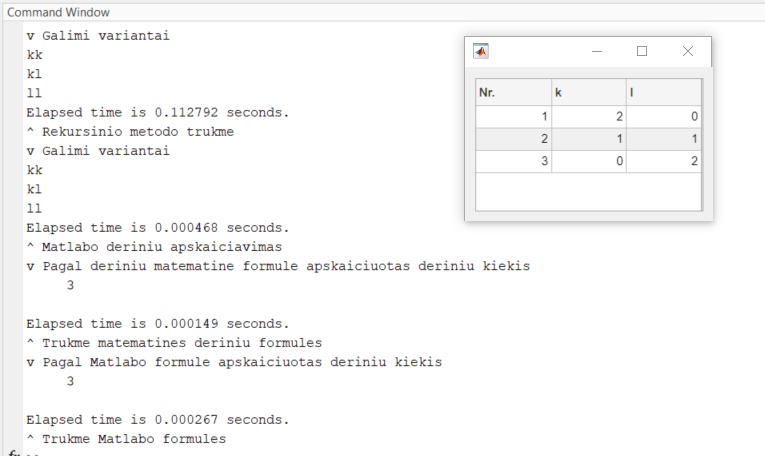
## Pirmas testas

Pirmojo pavyzdžio rezultatai pavaizduoti 1 pav., kai duoti šie duomenys:

k = 2;

elements = ["k", "l"];

Šis variantas turi dviem skirtingus elementais sudaryti derinius. Programos rezultatas:



**1 pav**. Dvejais skirtingais elementais sudaryti derinius

Iš gautų rezultatų, galime matyti, kad Matlab’o funkcijos suranda derinius greičiau, nei mano parašytoji rekursinė funkcija.

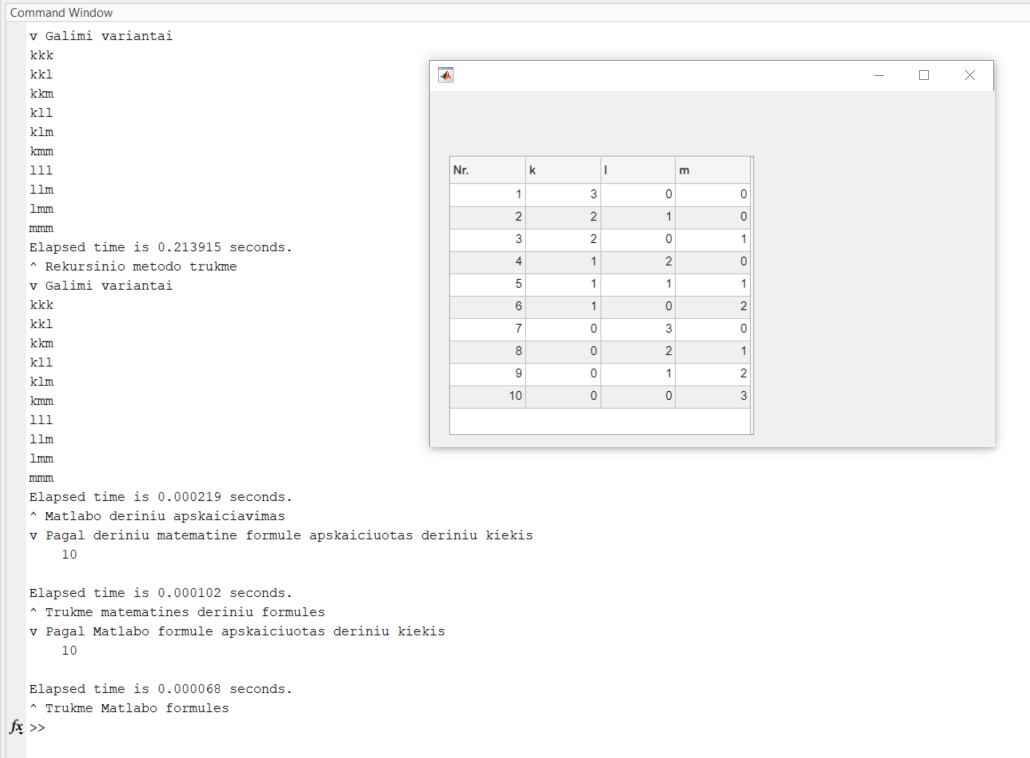
## Antras testas

Antrojo pavyzdžio rezultatai pavaizduoti 2 pav., kai duoti šie duomenys:

k = 3;

elements = ["k","l","m"];

Šis variantas turi su trimis skirtingais elementais sudaryti derinius. Programos rezultatas:



**2 pav**. Trimis skirtingais elementais sudaryti derinius

Iš gautų rezultatų, galime matyti, kad Matlab’o funkcijos suranda derinius greičiau, nei mano parašytoji rekursinė funkcija.

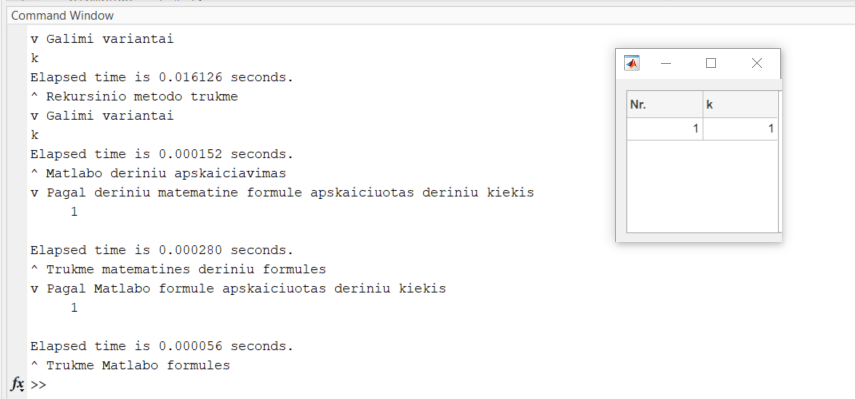
## Trečias testas

Antrojo pavyzdžio rezultatai pavaizduoti 3 pav., kai duoti šie duomenys:

k = 1;

elements = ["k"];

Šis variantas turi su vienu skirtingu elementu sudaryti derinius. Programos rezultatas:



**3 pav**. Vienu skirtingu elementu sudaryti derinius

Iš gautų rezultatų, galime matyti, kad Matlab’o funkcijos suranda derinius greičiau, nei mano parašytoji rekursinė funkcija.

# Išvados

Programa veikia teisingai. Mano parašytos funkcijos veikia truputį lėčiau, nei „Matlab“ programos esančios. Atlikus derinių suradimą, juos atspausdina ne tik „Command Window“ lange, bet ir yra atspausdinami lentelėje (grafinėje sąsajoje).

# Literatūros sąrašas

1. „Matlab“ dokumentacija <http://www.mathworks.se/help/index.html> (žiūrėta 2021-11-05)
2. „Diskrečiųjų struktūrų“ modulis „Moodle“ aplinkoje https://moodle.ktu.edu/course/view.php?id=39 (žiūrėta 2021-11-04)