

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS

Praktinė užduotis

**Tiesinio kodo C ir grandininio dekodavimo realizavimas
programinėje aplinkoje**

Atliko:

Programų sistemų 4 k. 1 gr. stud. Rokas Petrauskas

VILNIUS, 2023

Turiny

Realizuotos užduoties dalys	3
Naudotos bibliotekos	3
Užduoties atlikimo laikas	3
Programos paleidimo instrukcijos	3
Programos failai	3
Vartotojo sąsajos aprašymas	4
Padaryti programiniai sprendimai	5
Atlikti eksperimentai	5
Naudota literatūra	6

Realizuotos užduoties dalys

Klaidas taisantys kodai:

1. Baigtiniai kūnai **realizuoti**, aprašyti vektoriai, matricos, darbas su jais – daugyba, transponavimas, sudėtis, automatinis generavimas.
2. Kodavimas **realizuotas**.
3. Kanalas **realizuotas**.
4. Dekodavimas **realizuotas**.

Kita:

1. Ataskaita **įgyvendinta**.
2. Išeities tekstai: kodas komentuotas, stengiasi laikytis kodavimo standartų ir palaikyti bendrą stilių.
3. 1 scenarijus **įgyvendintas**, 2 ir 3 scenarijai pirminėje programos versijoje **neįgyvendinti**.

Naudotos bibliotekos

1. java.util.Scanner naudotas vartotojo įvesčiai.
2. java.util.Arrays naudotas palyginti dvejų matricų tapatumui.
3. java.util.Random naudotas kanalui simuliuoti, taip pat generuojančiai matricai G sukurti.

Užduoties atlikimo laikas

Viso darbą atlikti truko +-20 valandų.

Literatūros skaitymas bei užduoties aptarimo žiūrėjimas bei konspektavimas užtruko +- 4 valandas, ataskaitos pildymas truko +-2 valandas.

Projektavimo ir programavimo dalis truko +-7 valandas, tuo tarpu klaidų ieškojimui praleidau +-5 valandas.

Pakeitimai po atsiskaitymo truko +-2 valandas.

Programos paleidimo instrukcijos

Programa pateikiama archyve pavadinimu „A11 - JAVA - Rokas Petrauskas.7z“. Norint paleisti programą reikia:

1. Išarchyvuoti failo turinį (aplankas pavadinimu A11) pasirinktoje vietoje.
2. Atidaryti aplanką „A11“.
3. Atidaryti aplanką „out“.
4. Atidaryti aplanką „artifacts“.
5. Atidaryti aplanką „A11_jar“.
6. Atidaryti komandinę eilutę šioje direktorijoje.
7. Į komandinę eilutę suvesti tekstą: java -jar A11.jar
8. Programa paleidžiama, tolimesni veiksmai ir reikalingos vartotojo įvestys paaiškinamos programos veikimo metu.

Svarbu: kompiuteryje turi būti instaliuota JAVA vykdymo aplinka. Programuota ir testuota naudojant java 17.0.5 versiją.

Programos failai

Aplanke A11 pateiktos 5 java klasės.

1. Main.java – klasė skirta darbu

2. UI.java – klasė, apdorojanti didžiąją dalį programos išvedamo teksto vartotojui bei vartotojo duomenų įvedimą.
3. Matrix.java – klasė realizuojanti matricas bei vektorius ir darbą su jais - daugybą, transponavimą, sudėtį, automatinį generavimą.
4. LinearCode.java – klasė realizuojanti užkodavimo, dekodavimo algoritmus bei juose naudojamąs pagalbines funkcijas.
5. Channel.java – klasė realizuojanti kanalą.

Vartotojo sąsajos aprašymas

Atidarius programą gaunamas pranešimas „Įveskite koduojamo vektoriaus ilgį (kodo dimensiją)“. Vartotojas įveda kodo dimensiją (k) ir spaudžia Enter.

Gaunamas pranešimas „Įveskite kodo ilgį:“. Vartotojas įveda kodo ilgį (n) ir spaudžia Enter.

Jei įvedamas netinkamas kodo ilgis, gaunamas pranešimas „Kodo ilgis turi būti lygus arba didesnis už kodo dimensiją. Įveskite kodo ilgį:“. Vartotojas vėl įveda kodo dimensiją (k) ir spaudžia Enter. Tai kartojama kol k yra tinkamas.

Gaunamas pranešimas „Jei norite įvesi generuojančią matricą, įveskite simbolį "y":“. Jei vartotojas nori įvesti generuojančią matricą G pats įvedą simbolį y ir spaudžia Enter. Tada Parašo pirmos eilutės elementai iš kairės į dešinę po kiekvieno spaudžiant enter. Kai užpildoma pirmoji eilutė, vedama antroji, taip, kol užpildoma visa matrica. Pvz.: norime įvesti matricą G:

1, 0

0, 1

Vartotojas gavęs pranešimą „Jei norite įvesi generuojančią matricą, įveskite simbolį "y":“ spaudžia: y, Enter, gaunamas pranešimas „Įveskite matricos elementus, po kiekvieno elemento spauskite ENTER“. Vartotojas spaudžia: 1, Enter, 0, Enter, 0, Enter, 1, Enter

Jei vartotojas matricos įvesti nenori, gali spausti Enter arba bet kokį kitą tekstą ir Enter. Abejais atvejais vartotojui atspausdinama matrica G.

Gaunamas pranešimas „Pasirinkite programos darbo scenarijų įvesdami numerį:

1. Vektoriaus kodavimas
2. Teksto kodavimas
3. Paveikslėlio kodavimas
0. Išėiti iš programos“

Vartotojas paspaudžia skaičių, atitinkantį norimą scenarijų (šiuo metu įgyvendintas tik pirmasis) ir Enter.

Gaunamas pranešimas „Įveskite <k> bitų ilgio vektorius:“. Vartotojas įveda k bitų ilgio vektorius ir spaudžia Enter. Jei vektorius įvestas neteisingai, gaunamas pranešimas „Neteisingai įvestas vektorius. Prašome įvesti 5 bitų ilgio vektorius, kuris susideda tik iš 0 ir 1.“ Vartotojas prašomas iš naujo įvesti vektorius, tai kartojama kol įvedamas tinkamas vektorius. Tada prašoma įvesti kanalo klaidos tikimybę, tikimybė yra realusis skaičius intervale [0;1], kur dešimtainė pozicija skiriama kableliu. Programa atspausdina užkoduotą vektorius c, vektorius gautą iš kanalo, klaidų skaičių bei kodą gautą iš kanalo, kur kiekvienas pakeistas bitas rašomas tarp laužtinių skliaustų, pvz.: užkoduotame vektoriuje c = 101010010 sugadinta pirmoji pozicija, vektorius gautas iš kanalo r atspausdinamas kaip [0]01010010.

Vartotojo klausama ar jis nori iš naujo įvesti iš kanalo gautą vektorius, jei taip, vartotojas jį tiesiog įveda ir spaudžia Enter, jei ilgis ar formata netinka, prašoma tą daryti dar kartą. Jei vartotojas vektoriaus keisti nenori, tiesiog spaudžia Enter.

Galų gale atspausdinamas dekodotas vektorius (rezultatas gautas po dekodavimo funkcijos bei galutinai dekodotas vektorius).

Padaryti programiniai sprendimai

Kadangi pirminėje programos versijoje nerealizuoti nei antrasis, nei trečiasis scenarijai, reikalavimuose aprašytų programinių sprendimų priimti nereikėjo.

Atlikti eksperimentai

Eksperimente nustatomas klaidų taisymo efektyvumas keičiantis klaidos tikimybei. Šiam konkrečiam eksperimentui naudoti parametrai:

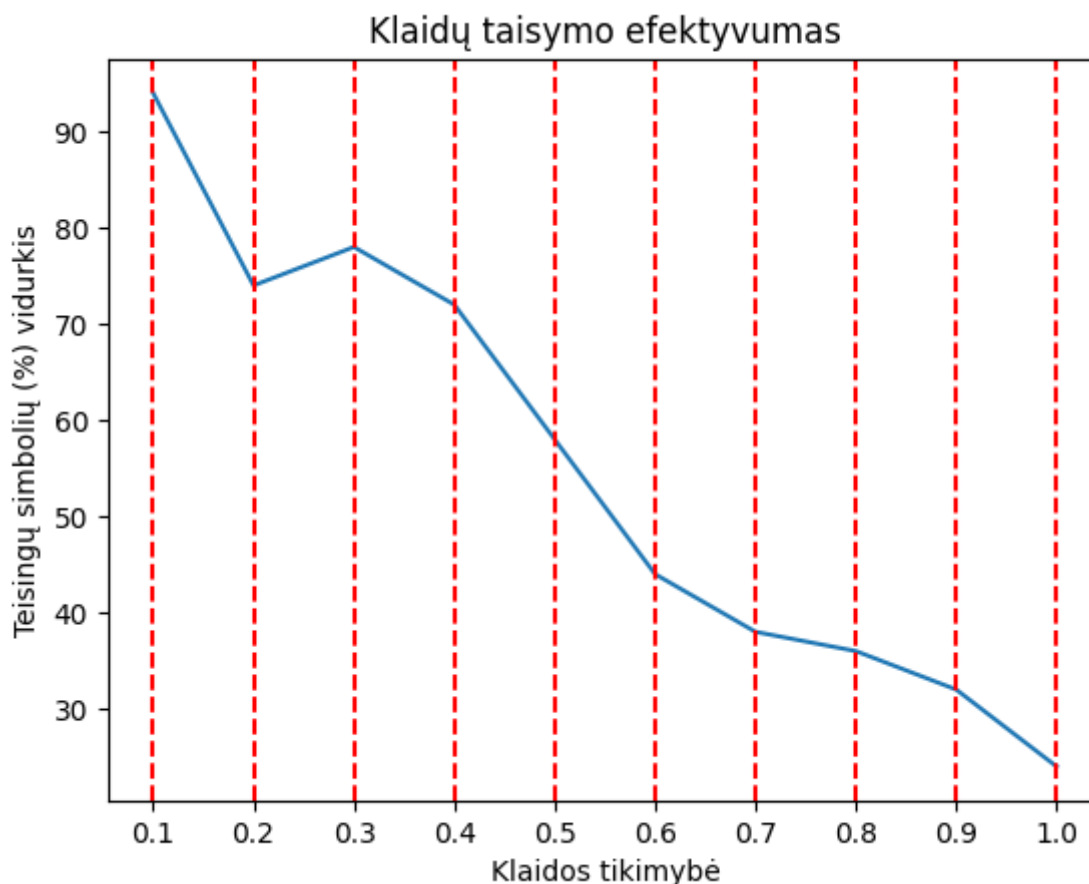
Kodo dimensija: 5

Kodo ilgis: 10

Generuojanti matrica: atsitiktinė vienetinė matrica.

Kodas: atsitiktinis 5 bitų ilgio vektorius.

Pav 1. Ir **Lentelė 1** matomi rezultatai naudojant aukščiau paminėtus parametrus. Kadangi įvestyje naudojama daug atsitiktinumo, rezultatai atspindi daug galimų skirtingų atvejų bei leidžia identifikuoti esminę tendenciją – didėjant klaidos tikimybei, mažėja klaidų taisymo efektyvumas. Leidžiant programą su skirtinga kodo dimensija ir kodo ilgiu, tendencija būtų tokia pati, tačiau klaidų taisymo efektyvumo mažėjimas būtų lėtesnis ar greitesnis priklausomai nuo naudojamų parametrų.



Pav. 1. Klaidų taisymo efektyvumas.

Lentelė 1. Klaidų taisymo efektyvumas.

Klaidos tikimybė	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
Efektyvumas (%)	100	100	80	100	40	20	20	20	40	20
	80	80	100	60	80	40	60	20	40	20
	100	100	100	100	80	60	20	20	40	40
	100	60	80	80	40	60	80	60	0	20
	100	100	80	60	60	60	40	40	40	20
	90	40	60	60	40	40	20	80	0	20
	100	100	80	80	40	60	60	20	60	20
	80	20	80	40	60	20	20	20	40	20
	90	80	60	80	80	60	20	40	0	40
	100	60	60	60	60	20	40	40	60	20
Teisingų simbolių (%) vidurkis	94	74	78	72	58	44	38	36	32	24

Naudota literatūra

S.A.Vanstone, P.C. van Oorschot. An introduction to error correcting codes with applications. Kluwer Academic Publishers, Boston, 1989

G. Skersys. Klaidas taisančių kodų teorija Paskaitų konspektai. VU MIF, 2021.