VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

PROGRAMŲ SISTEMŲ KATEDRA

**A11 Užduotis**

Ataskaita

Atliko:                             3 kurso 2 grupės studentas:

Audrius Kumpis

Darbo vadovas:  dr. Gintaras Skersys

Vilnius – 2021

Turinys

[1. Realizuotos dalys 3](#_Toc89792495)

[2. Trečiųjų šalių naudotos bibliotekos 3](#_Toc89792496)

[3. Užtrukęs laikas 3](#_Toc89792497)

[4. Programos paleidimas 3](#_Toc89792498)

[5. Tekstiniai failai ir kas juose realizuota 3](#_Toc89792499)

[6. Vartotojo sąsajos aprašymas su naudojimo pavyzdžiais: 4](#_Toc89792500)

[7. Padaryti programiniai sprendimai 6](#_Toc89792501)

[8. Atliktų eksperimentų aprašymas: 7](#_Toc89792502)

[9. Naudotos literatūros sąrašas 10](#_Toc89792503)

# Realizuotos dalys

Visi trys scenarijai yra realizuoti.

# Trečiųjų šalių naudotos bibliotekos

Nėra trečiųjų šalių bibliotekų.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Sritis | Literatūros skaitymas | Kodo veikimo aiškinimasis | Projektavimas | Programavimas | Klaidų ieškojimas | Klaidų taisymas | Ataskaitos ruošimas |
| Valandos | 5 | 8 | 2 | 18 | 5 | 2 | 5 |
| Išviso (h) | 45 |

# Užtrukęs laikas

# Programos paleidimas

Galima paleisti programą per IDE (buvo kuriama su IntelliJ), arba susikompiliuoti per komandinę eilutę. Visą kodą dar galima atrasti [GitHub](https://github.com/Audoryosa/coding-theory-linear-code) repositorijoje. Lokaliai galima pasileisti, turint Java savo kompiuteryje, pradinėje direktorijoje paleidus „task.jar“ aplikaciją su komanda “java -jar task.jar”.

# Tekstiniai failai ir kas juose realizuota

|  |  |
| --- | --- |
| **Failo pavadinimas** | **Aprašymas** |
| **Channel.java** | Klasė, skirta pranešimo iškraipymui su duota tikimybe. |
| **CodingUtils.java** | Pagalbinė klasė, susijusi su kodavimo teorijos metodais, tokiais kaip generuojančios matricos sukūrimas, visų įmanomų *n* ilgio vektorių generavimas. |
| **Decoder.java** | Klasė, skirta užkoduoto pranešimo dekodavimui. Joje taip pat yra apskaičiuojama poaibių lentelė, bei poaibių lyderių žemėlapis (*Map*). |
| **Encoder.java** | Klasė, skirta pranešimo užkodavimui pagal duotą generuojančią matricą. |
| **MatrixCalculationUtils.java** | Klasė, atsakinga už visas reikalingas operacijas su matricomis, pvz. transponavimas, matricų daugyba, vienetinės matricos generavimas. |
| **Main.java** | Pagrindinė klasė, kuri yra programos pradinis taškas. Joje yra konsolinė vartotojo sąsaja. Čia taip pat surenkama visa pradinė informacija, kaip generuojančios matricos dimensijos, klaidos tikimybė. |

# Vartotojo sąsajos aprašymas su naudojimo pavyzdžiais:

Paleidus programą, vartotojo yra paprašoma suvesti pradinius duomenis: pranešimo ilgį *k* ir dimensiją *n*.

Graphical user interface, text

Description automatically generated

Tuomet vartotojas gali pasirinkti, ar pats nori suvesti generuojančią matricą, ar leisti tai atlikti programai. Jei pasirenkama, jog vartotojas ją suves pats, programa paprašo įvesti *k* eilučių ir *n* stulpelių. Įvedus, programa atspausdina gautą matricą.

Text

Description automatically generated

Jei vartotojas nori leisti programai sugeneruoti matricą, jam nieko daryti nereikia. Pagal gautus parametrus, programa pati sukuria G matricą, pavidalu (I | A), kur I - vienetinė matrica, A – bet kokia matrica.

A picture containing text

Description automatically generated

Tuomet, vartotojo paprašoma įvesti kanalo klaidos tikimybę, nuo 0-1 imtinai.



Įvedus klaidos tikimybę, visi reikalingi parametrai jau yra išsaugoti. Vartotojas gali pasirinkti:

1. Dvejetainį kodą
2. Tekstą
3. Paveiksliuką.

Text

Description automatically generated

Pasirinkus dvejetainio kodo opciją, vartotojas įveda *k* ilgio dvejetainę seką. Programa parodo, kaip ji buvo užkoduota, ir kaip atrodo iškraipytas užkoduotas pranešimas. Jei norima, galima pakeisti užkoduotą iškraipytą pranešimą. Jei ne, toliau vyksta dekodavimas, ir parodomi rezultatai, kartu su neteisingai dekoduotų bitų skaičiumi, bet klaidų procentu.

Text

Description automatically generated

Panaši eiga vyksta pasirinkus teksto pasirinkimą:

Text

Description automatically generated

Pasirinkus paveikslėlio pasirinkimą, vartotojas įveda pilną kelią iki paveikslėlio. Tuomet programa ji užkoduoja, iškraipo, ir dekoduoja. Iškraipytas, dekoduotas bei originalas paveikslėlis susikuria programos lokacijoje. Parodomas dekodavimo klaidų procentas.

Text

Description automatically generated

# Padaryti programiniai sprendimai

1. Teksto ir paveikslėlių kodavimas/dekodavimas. Kadangi šie elementai turi labai daug bitų, tai algoritmas negali vienu metu jų visų imti ir dekoduoti. Kad ši problema būtų išspręsta, visi bitai yra išskaidomi į *k* ilgio masyvus, ir su visais jais atskirai vykdomi visi veiksmai. Tai labai pagreitino programos veikimą.
2. Nuskaitant paveikslėlio baitus ir konvertuojant juos į dvejetainį pavidalą, dažnai jie tapdavo trumpesni nei 8 bitų. Šiai problemai išspręsti, nuskaitoma teksto eilutė, kurioje yra visi bitai, kas 8 pozicijas, ir pridedama tiek nulių priekyje, kiek jų trūksta.
3. Po bitų skaidymo masyvais, kai kurie bitai tampa „nukąsti“ ir su jais paskiau negalima atlikti matricų operacijų. Mano sprendimas: prie jų galo pridedu tiek 0, kiek trūkstą.
4. Standartinės poaibių lentelės kūrimas. Tai buvo dalis, prie kurios užtrukau ilgiausiai. Mano sprendimas: sugeneruoju visus įmanomus *x* ilgio poaibius, ir apskaičiuoju jų sindromą bei sindromo svorį. Tada rezultatus sugrupuoju pagal sindromus. Iš kiekvienos sindromo grupės paimu tokį poaibį, kurio svoris yra mažiausias. Jį priskiriu poaibio lyderiu, ir su *java.util.Map* pagalba, susieju jį su sindromu bei jo svoriu.
5. Skaidant bitus gabalais, kiekvienas gabalas yra prasiunčiamas pro iškraipymo kanalą.

# Atliktų eksperimentų aprašymas:

1. **Teksto siuntimas kanalu užkodavus ir dekodavus bei teksto siuntimas kanalu, kai klaidos tikimybė yra 0.9.**

A picture containing flower, plant

Description automatically generatedOriginalas:

A picture containing plant, palm, fabric

Description automatically generatedIškreiptas:

A picture containing plant, flower, fabric

Description automatically generatedDekoduotas:

Algoritmas geriausiai ištaiso klaidas, kai *k=1*.

1. **Teksto siuntimas kanalu užkodavus ir dekodavus bei teksto siuntimas kanalu, kai klaidos tikimybė yra 0.05.**

* **Kai k = 5, n = 1, o generuojanti matrica yra (1 1 1 1 1)**

**Pradinis tekstas:** Kodavimo teorija yra labai sunkus, bet idomus modulis. Rekomenduoju!

**Tik pro kanalą prasiųstas tekstas:** KodavéíO teoò)j! y²ã lafai sufkus, bdt idG)u{ mo$ulIq& ReiomEndugju!

**Užkoduotas ir dekoduotas tekstas:** Kodavimo teorija yra labai sunkus, bet idomus modulis. Rekomendugju!

Text

Description automatically generated

* **Kai k = 10, n = 8**

**Pradinis tekstas:** Kodavimo teorija yra labai sunkus, bet idomus modulis. Rekomenduoju!

**Tik pro kanalą prasiųstas tekstas:** kdAvimo tåoriJa ýpa larai(senkuw, \*Et iäoíus0mofulmc. RejomanduojU!

**Užkoduotas ir dekoduotas tekstas:** KdAvIoo tåorija Ýra larai senkuW "et iäoíus0modulMc. RebomAnduoju!

Text

Description automatically generated

* **Kai k = 10, n = 1**

**Pradinis tekstas:** Kodavimo teorija yra labai sunkus, bet idomus modulis. Rekomenduoju!

**Tik pro kanalą prasiųstas tekstas:**   
oäavimo teNsIha {sa Ìabqi`óunjuW,(Fe| mdoOtc mo%umHs.,Ra«gmunduo\*u!

**Užkoduotas ir dekoduotas tekstas:** Kodavimo teorija0yra labai sunkus, bet idomus modulis. RekOmenduoju!

Text

Description automatically generated

1. **A picture containing text, electronics, keyboard, close

   Description automatically generatedKlaidų kiekio priklausomybė nuo klaidų skaičiaus vektoriuje, kai n = 7 , k = 4, G matrica =**

Iš rezultatų galima pamatyti, jog kodas geriausiai taiso klaidas, jei jų yra nedaug. Kitu atveju, ištaisoma maždaug pusė klaidų. Kartais, ištaisomas didelis skaičius klaidų. Taigi, šis kodas geriausiai taiso nedidelį kiekį (3-4) klaidų, kitu atveju, jis tampa nepatikimas.

1. **Klaidų kiekio priklausomybė nuo klaidų skaičiaus vektoriuje, kai n = 7 , k = 1, G matrica =**



Kaip generuojanti matrica turi tik vieną stulpelį, klaidos ištaisomos yra daug efektyviau. Beveik kiekvieną vektorių ištaisė su iki 15% blogo dekodavimo skaičiumi. Išvada: šis kodavimo algoritmas geriausiai veikia, kai generuojanti matrica turi tik vieną stulpelį.

# Naudotos literatūros sąrašas

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

Klaidas taisančių kodų teorija. Paskaitų konspektai. Gintaras Skersys, 2021.