

Klaidas taisantys kodai

Užduoties A13 ataskaita

Golėjaus kodas

Gabija Liorentaitė
Informatika
4 kursas, 2 grupė

1. Užduoties dalys.....	2
2. Bibliotekos.....	3
3. Programos paleidimas.....	3
4. Programos failai.....	3
5. Naudotojo sąsajos aprašymas.....	4
6. Programiniai sprendimai.....	4
7. Eksperimentai.....	6
8. Literatūros sąrašas.....	7

1. Užduoties dalys

Visos užduoties dalys realizuotos **Python** kalba.

Bendroji dalys (iš užduoties aprašymo):

Duotas Goléjaus kodas C_{23} virš F_q . Modeliuoti jo veikimą:

- užkoduoti informaciją,
- siųsti ją nepatikimu kanalu, t.y. su duota tikimybė p atsitiktinai joje padaryti klaidų
- dekoduoti, naudojant nurodytą dekodavimo algoritmą iš duotos literatūros.

Programos vartotojas turi turėti galimybę nurodyti konkretų kūną (Goléjaus atveju q gali įgyti tik vieną reikšmę, q=2), konkretų kodą (šiuo atveju nėra) ir klaidos tikimybę p.

Programos vartotojo požiūriu turėtų būti realizuoti trys scenarijai:

1. Vartotojas užrašo programos nurodyto ilgio vektorių iš kūno F_q elementų. Programa patikrina, ar vektoriaus ilgis korektiškas, jį užkoduojant kodu C, parodo užkoduotą vektorių, siunčia jį kanalu, parodo iš kanalo išėjusį vektorių, praneša, kiek ir kuriose pozicijose įvyko klaidų, dekoduojant gautą vektorių ir parodo rezultatą. Vartotojas turi turėti galimybę prieš dekodavimą redaguoti iš kanalo išėjusį vektorių, kad galėtų pats nurodyti klaidas ten, kur jam reikia, ir tiek, kiek reikia.
2. Vartotojas užrašo tekstą. Programa suskaido duotą tekštą į reikiama ilgio vektorius iš kūno F_q elementų. Kiekvieną vektorių siunčia kanalu nenaudojant kodo, iš gautų vektorių atgamina tekštą ir jį parodo. Kiekvieną vektorių užkoduojant, siunčia kanalu, dekoduojant. Iš gautų vektorių atgamina tekštą ir jį parodo.
3. Vartotojas nurodo paveiksliuką. Programa atidaro jį ir parodo. Suskaido duotą paveiksliuką į reikiama ilgio vektorius iš kūno F_q elementų. vektorių siunčia kanalu nenaudodama kodo, iš gautų vektorių atgamina paveiksliuką ir jį parodo. Kiekvieną vektorių užkoduojant, siunčia kanalu, dekoduojant. Iš gautų vektorių atgamina paveiksliuką ir jį parodo.

2. Bibliotekos

A. **Random**

Bazinė Python biblioteka. Naudojama atsitiktinių skaičių generavimui kanalui iškraipyti.

B. **Pillow (PIL - Python Imaging Library)**:

Naudojama paveikslėlių apdorojimui.

C. **NumPy (Numerical Python)**:

Naudojama tik su Pillow biblioteka paveikslėlių formatavimui.

D. **Matplotlib.pyplot**

Naudojama eksperimentų grafiniam atvaizdavimui.

3. Programos paleidimas

Programai paleisti reikia būti jisidiebus Python aplinką (šiam kodui buvo naudota 3.14 versija). Programą galima paleisti per terminalą iš to aplanko, kur yra visi failai išvardyti 4.Programos failai. Terminale reikia išrašyti *py main.py* ir programa bus paleista.

4. Programos failai

- 4.1. **main.py** – Pagrindinis failas su vartotojo sąsaja, per kurį galima pasiekti visas tris dalis: vieno vektoriaus, teksto ir paveikslėlio apdorojimą.
- 4.2. **data.py** - Konstantų failas su Goléjaus kodo B, G, H, I matricomis.
- 4.3. **functions.py** - Realizuoti matricų veiksmai.
- 4.4. **encoding.py** – Goléjaus kodo užkodavimo algoritmas ir vektoriaus siuntimas nepatikimu kanalu.
- 4.5. **decoding.py** – Goléjaus kodo dekodavimo algoritmas.
- 4.6. **formatting.py** – Funkcijos tekstu ir paveikslėlių konvertavimui į dvejetainę formą ir atgal.
- 4.7. **vector_processor.py** – Vektoriaus siuntimas kanalu be kodo ir su kodu. Taip pat parodo kiek ir kuriose pozicijose įvyko klaidų, bei duoda galimybę pakeisti išeinančią iš kanalo vektorių.
- 4.8. **text_processor.py** – Teksto siuntimas kanalu be kodo ir su kodu.
- 4.9. **image_processor.py** – Paveikslėlio siuntimas kanalu be kodo ir su kodu.
- 4.10. **experiment.py** – Papildomas failas, skirtas testuoti vektoriaus kodavimą su skirtingomis klaidos tikimybėmis.

5. Naudotojo sasajos aprašymas

Paleidus programą, naudotojui terminale parodomos pagrindinis meniu, kuriame galima pasirinkti vieną iš keturių funkcijų:

1. vektoriaus apdorojimą
2. teksto apdorojimą
3. paveikslėlio apdorojimą
4. išėjimą iš programos

Naudotojas turi įvesti skaičių nuo 1 iki 4. Jei įvedamas kitas simbolis, programa parodo klaidos pranešimą ir prašo pasirinkti iš naujo.

Pasirinkus vektoriaus apdorojimą (1) naudotojas turi įvesti 12 bitų ilgio dvejetainį vektorių (galima su tarpais arba kableliais). Jei įvedami kitokie simboliai arba vektoriaus ilgis ne 12, programa išveda klaidos pranešimą.

Po to naudotojas turi įvesti klaidos tikimybę p nuo 0.0 iki 1.0. Vektorius automatiškai užkoduojamas, perduodamas per kanalą ir parodoma: įvestas vektorius, išsiųstas vektorius ir gautas vektorius (su klaidomis), bei klaidų skaičius ir klaidų pozicijos.

Naudotojas gali pasirinkti, ar nori rankiniu būdu pataisyti gautą 23 bitų vektorių (galima su tarpais arba ne). Jei įvedamas neteisingo formato vektorius, parodoma klaida. Po visų veiksmų programa išveda dekoduotą vektorių ir pasako ar pavyko ji teisingai dekoduoti.

Pasirinkus teksto apdorojimą (2) naudotojas terminale įveda tekštą. Įvedimas baigiamas du kartus paspaudus „Enter“. Naudotojas taip pat turi įvesti klaidos tikimybę p nuo 0.0 iki 1.0. Programa pirmiai perduoda tekštą be kodavimo ir parodo, kaip jis išsikraipo, o po to pakartoja procesą su Golėjaus kodu ir parodo, koks tekstas atkuriamas dekodavus.

Pasirinkus paveikslėlio apdorojimą (3) naudotojas turi įvesti paveikslėlio failo pavadinimą. Taip pat prašoma įvesti klaidos tikimybę p nuo 0.0 iki 1.0. Programa atidaro 3 paveikslėlius: originalų paveikslėlį, iškreiptą paveikslėlį be kodavimo ir paveikslėlį panaudojus Golėjaus kodą. Programa išsaugo du paveikslėlius: be kodavimo ir su kodavimu. Abu failai išsaugomi tame pačiame aplanke kaip „image_no_code.bmp“ ir „image_with_code.bmp“.

Po vektoriaus, teksto ar paveikslėlių apdorojimo programa vėl gržta į pagrindinį meniu.

6. Programiniai sprendimai

Teksto pavertimas į dvejetainius vektorius

Pirmiausia naudotojo įvestas tekstas yra paverčiamas į dvejetainę formą. Kiekvienas simbolis yra konvertuojamas pagal ASCII kodą į dvejetainį skaičių. Tam naudojama funkcija *stringToBinary*. Ji grąžina visų simbolių dvejetaines reikšmes, atskirtas tarpais.

Dvejetainės sekos padalinimas į 12 dydžio vektorius

Kadangi kiekvienas vektorius turi būti 12 ilgio kodavimui, programa kiekvieną gautą dvejetainį vektorių išplečia nuliais iki 12 bitų ilgio. Tai atliekama funkcijoje *splitBinary*.

Jeigu simbolio dvejetainis kodas yra trumpesnis (pvz., 7 ASCII bitai), trūkstami bitai pridedami gale. Programa taip pat išsaugo, kiek nulių buvo kiekvienam vektoriui pridėta, nes vėliau atkuriant tekštą juos reikia pašalinti.

Vektorių siuntimas kanalu

Programa leidžia siųsti vektorius per kanalą. Kanalus imituojamas paprastai – perduodamo vektoriaus kiekvienas bitas gali būti apverstas su paduota galimybe p. Tam naudojama funkcija *transmit*.

Gauto vektoriaus atkūrimas į tekštą

Gavus vektorius iš kanalo, jie yra atkoduojami atgal į ASCII simbolius funkcijoje *binaryToString*.

Jei gautas vektorius yra 23 bitų (be kodavimo teksto parodymui), paimami tik pirmi 12 bitų. Iš šių 12 bitų pašalinami papildyti nuliai. Bitų seka paverčiama į sveiką skaičių (ASCII kodą). Tada skaičiai paverčiami į simbolius ir sujungiami į tekštą.

Vektorių formavimas paveikslėliui

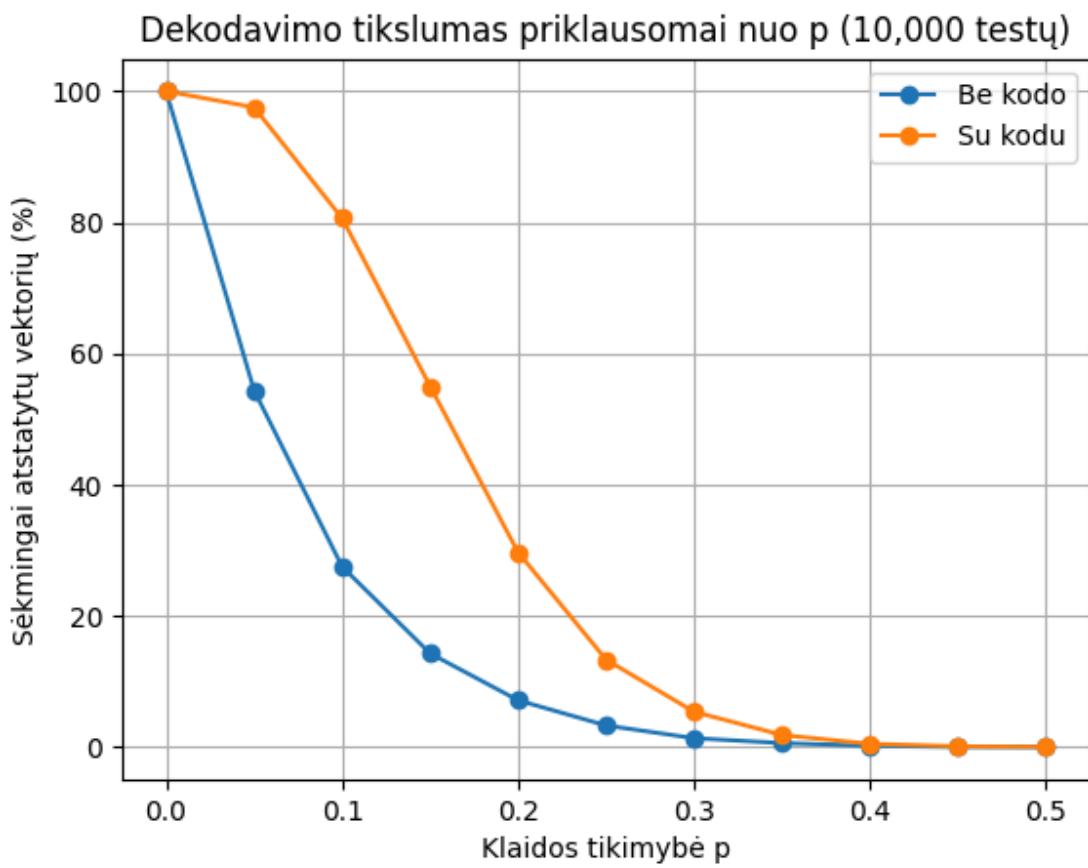
Gauti dvejetainiai vektoriai pirmiausia paverčiami į sveikus skaičius, kurie atitinka pikselių pilkos spalvos intensyvumą (0–255). Šiuo metu vektoriai yra vienmatyje masyve.

Tada vienmatis sąrašas perkuriamas į dvimatių masyvą pagal paveikslėlio aukštį ir plotį, kur kiekviena eilutė atitinka paveikslėlio eilutę. Gautas dvimatis masyvas perduodamas *Pillow* bibliotekos funkcijai *fromarray*, kad būtų galima sukurti paveikslėlį, jį išsaugoti ar parodyti ekrane.

7. Eksperimentai

7.1. Vektoriai

Buvo testuoti atsitiktinai sugeneruoti 12 ilgio vektorių su klaidos tikimybėmis nuo 0 iki 0.5. Klaidos tikimybės didėjo 0.05 intervalu ir kiekvienai tikimybei buvo generuoti po 10 tūkstančių vektorių, lyginant rezultatus be kodo ir su Golējaus kodu.



7.2. Nuotrauka

Testuojama ta pati nuotrauka 3 kartus su skirtingomis klaidos tikimybėmis.
 Matuojama be kodavimo ir su Goléjaus kodavimo proceso trukmė sekundėmis.
 Nuotraukos dydis 128x128

Klaidos tikimybė p	Procesavimo laikas be kodavimo (s.)	Procesavimo laikas su kodavimu (s.)
0.0	0.15	3.93
0.1	0.11	3.76
0.2	0.11	3.86

0.3	0.12	3.94
0.4	0.11	3.95
0.5	0.11	4.12
0.6	0.16	4.08
0.7	0.11	3.82
0.8	0.15	4.01
0.9	0.13	4.31
1.0	0.12	3.81

8. Literatūros sąrašas

- 8.1. D.G.Hoffman, D.A.Leonard, C.C.Lindner, K.T.Phelps, C.A.Rodger, J.R.Wall. Coding Theory: The Essentials. Dekker, New York, 1991.
- 8.2. Gintaras Skersys Klaidas taisančių kodų teorija Paskaitų konspektai. VU MIF, 2025
- 8.3. Mistral AI - eksperimento atvaizdavimui grafiškai.