

## 1. Užduotis A5

- Kodavimas: Rydo-Miulerio (Reed-Muller) kodas RM(1,m)
- Dekodavimas: dekodavimo algoritmas, naudojantis greitąją Hadamardo transformaciją (Fast Hadamard Transform)
- Parametrai:  $q=2$ ,  $m$

## 2. Realizuotos dalys

Darbe realizuotos visos privalomos dalys.

## 3. Darbo trukmė

- Literatūros skaitymui ir kodo veikimo aiškinimui: apie 4h
- Vartotojo sąsajos projektavimui ir įgyvendinimui: apie 3h
- Programos projektavimui: apie 2h
- Programavimui: apie 15h
- Klaidų ieškojimui ir taisymui: apie 5h
- Programinio kodo pertvarkymui, tobulinimui, dokumentavimui: apie 3h
- Ataskaitos ruošimui ir eksperimentų atlikimui: apie 3h

Viso: apie 35h.

## 4. Programos paleidimas

Programa paleidžiama **Reed-Muller.exe** failu.

## 5. Programos tekstų failų turinys

*Program.cs* - Programos paleidimas

- *Coding* aplankas
  - *Channel.cs* - Realizuotas pranešimo siuntimas kanalu
  - *Decoder.cs* - Realizuotas vektorių dekodavimas
  - *Encoder.cs* - Realizuotas vektorių kodavimas
- *Models* aplankas
  - *GeneratorMatrix.cs* - Realizuotas generuojančių matricų sukūrimas ir užpildymas
  - *HadamardTransformMatrix.cs* - Realizuotas vienetinių matricų ir Kronekerio matricų generavimas ir užpildymas
  - *Vector.cs* - Baigtinio kūno - vektoriaus - klasė

- *Utils* aplankas
  - *ConversionUtils.cs* - Realizuotos pagalbinės funkcijos atliekant tipų konversijas (iš tekstinės eilutės į int masyvą, iš paveikslėlio į binarinį masyvą ir t.t.)
  - *MatrixUtils.cs* - Realizuota vektoriaus ir matricos daugyba
  - *Utils.cs* - Realizuota pagalbinė funkcija sumažinti (sukompresuoti) pateiktus paveikslėlius
- *Views* aplankas
  - *ConfigurationForm.cs* - Realizuotas pradinis programos langas, kuriame įvedamas parametras m ir išskviečiamos funkcijos, generuojančios reikiamas matricas
  - *ImageView.cs* - Realizuota trečiojo scenarijaus (paveikslėlio siuntimo tinklu) vartotojo sąsaja
  - *MainView.cs* - Vartotojo sąsaja, kurioje realizuojamas scenarijaus pasirinkimas
  - *ProbabilityPanel.cs* - Vartotojo sąsajos elementas klaidų tikimybei įvesti
  - *TextView.cs* - Realizuota antrojo scenarijaus (teksto siuntimo tinklu) vartotojo sąsaja
  - *VectorView.cs* - Realizuota pirmojo scenarijaus (vektoriaus siuntimo tinklu) vartotojo sąsaja
- *img* aplankas
  - Šiame aplanke patalpinti testavimui tinkamo BMP formato paveikslėliai, kuriuos galima panaudoti tikrinant trečiojo scenarijaus veikimą.

## 6. Vartotojo sąsajos aprašymas

- **Pradinis (konfigūracijos) langas.** Šiame lange reikia įvesti kodo parametą m. Šio parametro reikšmė gali būti sveikasis skaičius nuo 1 iki 12 (įskaitant). Šis langas neleidžia įvesti mažesnės nei 1 reikšmės; reikšmės įvestas su kableliu, nukerta iki sveiko skaičiaus. Įvedus reikšmę, didesnę nei 12, programa nustoja veikti kai atliekamas (de)kodavimas, nes nebeužtenka sistemos atminties, ir parodo klaidos pranešimą.

Įvedus m ir paspaudus *Continue* atidaromas scenarijaus pasirinkimo langas.

- **Scenarijaus pasirinkimo lange** galima pasirinkti, ką norėsime koduoti ir siųsti kanalu - vektorių, tekstinį pranešimą ar paveikslėlį.
- **Vektoriaus kodavimo langas.** Šis langas atidaromas scenarijaus pasirinkimo lange paspaudus *Vector*. Pradžioje lange matome tik vieną įvedimo lauką ir mygtuką. Šiame lauke reikia įvesti nurodyto ilgio vektorių (sudarytą iš skaitmenų 0 arba 1). Įvedus netinkamą vektorių ir paspaudus *Encode* parodomas klaidos pranešimas. Įvedus tinkamą vektorių, jis užkoduojamas ir parodomas naujame atsiradusiame lange. Taip pat atsiranda klaidų tikimybės p pasirinkimo laukas.

Pasirinkus tikimybę (nuo 0 iki 1, skaičiai įvedami su skiriamuoju kableliu arba tašku) ir paspaudus *Send* mygtuką, imituojamas vektoriaus siuntimas kanalu su nurodyta tikimybe p. Iš kanalo gautas vektorius parodomas naujame tekstiniame lauke. Taip pat parodoma, kuriose užkoduoto vektoriaus vietose buvo padarytos klaidos. Iš kanalo gautą vektorių galima redaguoti (jo ilgis turi išlikti toks pats ir jis turi būti sudarytas iš 1 arba 0).

Paspaudus *Decode* iš kanalo gauto vektoriaus lauke esantis vektorius dekoduojamas ir parodomas naujame lauke.

- **Teksto kodavimo langas.** Šis langas atidaromas scenarijaus pasirinkimo lange paspaudus *Text*. Pirmame lauke prašoma įvesti tekstą, kurį norėsime siųsti tinklu. Tekstas gali būti kelių eilučių ilgio, bet, pageidautina, be lietuviškų raidžių (jos dekodavus būna prarastos). Įvedus tekstą ir įvedus klaidos tikimybę  $p$ , bei paspaudus *Send*, imituojamas teksto siuntimas kanalu dviem būdais. Pirmu būdu tekstas yra siunčiamas kanalu nekoduotas, antru būdu tekstas yra skaidomas reikiamo ilgio vektoriais, jie užkoduojami, siunčiami tinklu ir tada dekoduojami. Abiem būdais gautos žinutės pavaizduojamos atitinkamose lango vietose.
- **Teksto kodavimo langas.** Šis langas atidaromas scenarijaus pasirinkimo lange paspaudus *Image*. Pradžioje lange matomas vienas mygtukas - įkelti paveikslėlį. Paveikslėlį reikia įkelti BMP formatu. Kadangi didelių paveikslėlių programa sukompresuoja, rekomenduojama naudoti 1bpp (juodai-baltus) BMP paveikslėlius, mažesnius nei 300x200. Tokie paveikslėliai nebūna kompresuojami ir rezultatas pasimato geriausiai. Įkėlus paveikslėlį ir pasirinkus klaidos tikimybę  $p$ , bei paspaudus *Send*, imituojamas paveikslėlių siuntimas tinklu dviem būdais - vienu būdu paveikslėlis siunčiamas neužkoduotas, kitu būdu - paveikslėlis suskaidomas į atitinkamo ilgio vektorius, kurie užkoduojami, o persiuntus tinklu, dekoduojami. Gauti rezultatai parodomi šiame lange.

## 7. Padaryti programiniai sprendimai

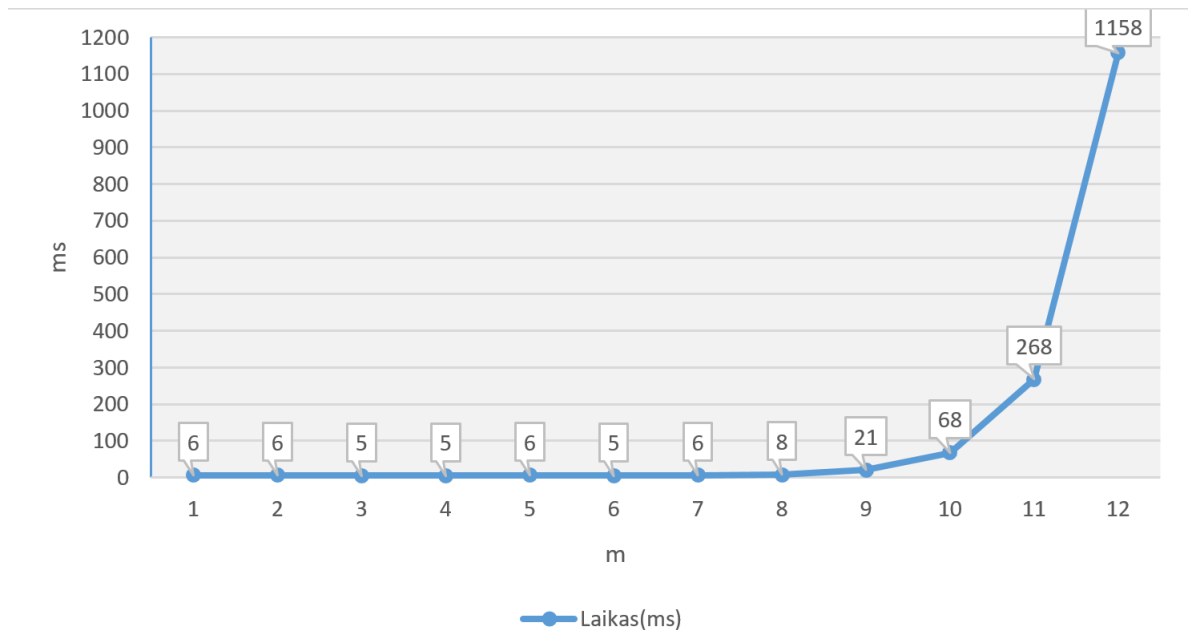
1. Teksto siuntimas tinklu. Siunčiant tekstą tinklu, kiekvienam vektoriaus bitui yra sugeneruojamas atsitiktinis skaičius. Jei šis skaičius mažesnis už klaidos tikimybę  $p$ , bitas invertuojamas (t.y. iš 0 pakeičiamas į 1 ir atvirkščiai).
2. Matricų generavimas. Generuojančios, vienetinės ir Krionikerio matricos generuojamos iškart įvedus parametrą  $m$ . Vienetinės matricos generuojamos tik dvejetainiams iki  $m-1$ . Sugeneruotos matricos saugomos programos atmintyje iki kol įvedamas naujas  $m$  - tada jos yra perskaičiuojamos iš naujo.
3. Teksto kodavimas. Įvedus tekstą, jis užkoduojamas naudojant ASCII koduotę ir paverčiamas bitų seka - dideliu vektoriumi. Šis vektorius, prieš jį koduojant, suskaidomas ilgio  $m+1$  vektoriais. Jeigu paskutinio vektoriaus ilgis gaunasi mažesnis nei  $m+1$ , prirašomi reikiami '0' bitai jo gale.
4. Teksto dekodavimas. Dekodavimu metu papildomi bitai yra pašalinami, dekodavimui naudojama ASCII koduotė.
5. Paveikslėlio siuntimas. Programa priima tik BMP formato paveikslėlius. Jeigu įkeltas paveikslėlis yra ne 1bpp pikselių formato ir didesnis 300x300, jis sumažinamas ir suspaudžiamas. Toks paveikslėlis praranda savo rezoliuciją ir ryškumą, tačiau užkertamas kelias programai apdoroti didelį paveikslėlį valandas laiko. Tad šis sprendimas leidžia naudoti įvairius BMP paveikslėlius, bet turi apribojimus norint išgauti geriausią rezultatą.  
Įkeltas paveikslėlis yra konvertuojamas į bitus - binarinį masyvą, kuris, kodavimo metu, yra skaidomas  $m+1$  ilgio vektoriais. Paveikslėlio *headeris* - 54 bitai - nėra koduojamas ir siunčiamas tinklu - jis nukerpami nuo siunčiamo vektoriaus ir apjungiamas jį dekodavus.
6. Kelių gijų naudojimas. Teksto ir paveikslėlio kodavimo scenarijų languose, paspaudus *Send* mygtuką, sukurama nauja gija, kurioje atliekami apdorojimo procesai - tokiu būdu neblokuojama pagrindinė gija ir galima toliau dirbti su programa. Tiesa, *Send* mygtukas tampa neaktyvus iki kol kodavimo gija nepabaigia savo procesų.

## 8. Atlikti eksperimentai

### 8.1. Vektoriaus (de)kodavimo laiko priklausomybė nuo $m$

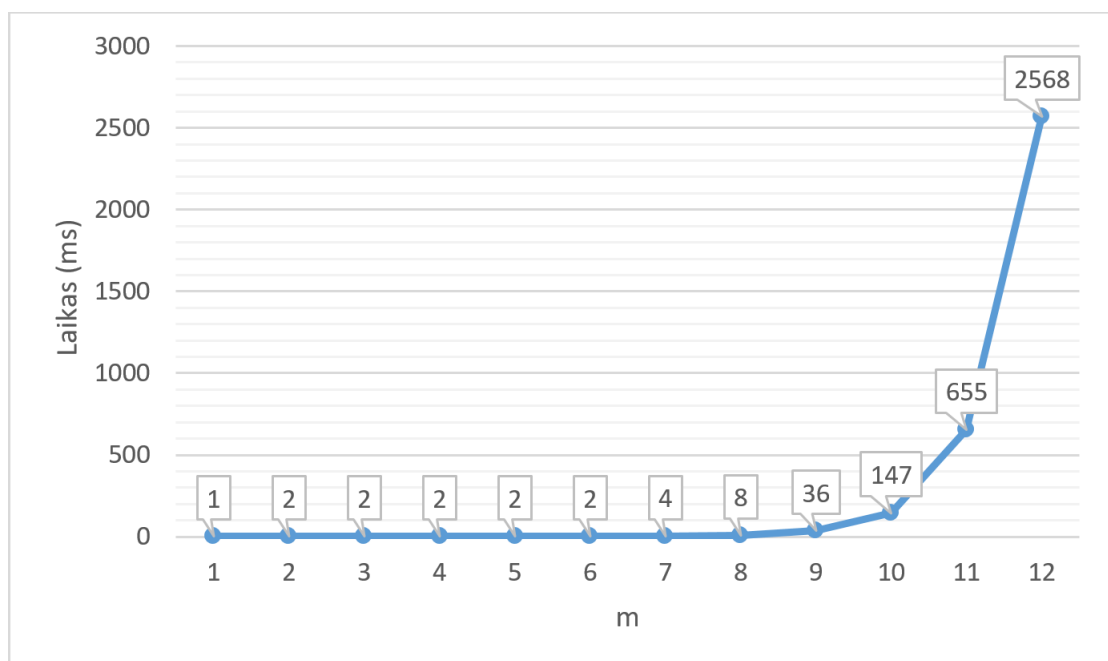
Šio bandymo metu buvo fiksuotas laikas, kiek užtruko užkoduoti ir dekoduoti iš tinklo gautą vektorių naudojant parametą  $m$  ( $1 \leq m \leq 12$ ). Užkodavimo ir dekodavimo laikai buvo fiksuojami atskirai ir vėliau sudedami į bendrą sumą.

Galima pastebėti, kad kai  $m=12$ , vykdymo trukmė stipriai išauga - taip yra todėl, kad tenka apdoroti itin didelių matmenų matricas.



### 8.2. Matricų generavimo laiko priklausomybė nuo $m$

Į pirmąjį panašus eksperimentas buvo atliktas ir tikrinant matricų (generuojančių, vienetinių ir Kronekerio) generavimo trukmę programos paleidimo pradžioje, kai įvedamas parametras  $m$ .



### 8.3. Ištaisomų klaidų kiekio priklausomybė nuo m

Šio eksperimento metu visiems m nuo 1 iki 8, buvo kanalu siunčiamas vektorius, sudarytas vien iš '1'. Tada užkoduotame vektoriuje, pradedant nuo jo galo, kiekvienas '0' buvo keičiamas į '1' ir tikrinama, su kiek tokių padarytų klaidų, vektorius vis dar bus dekoduoamas teisingai į pradinį vektorių. Atliekant eksperimentą buvo gauti tokie rezultatai:

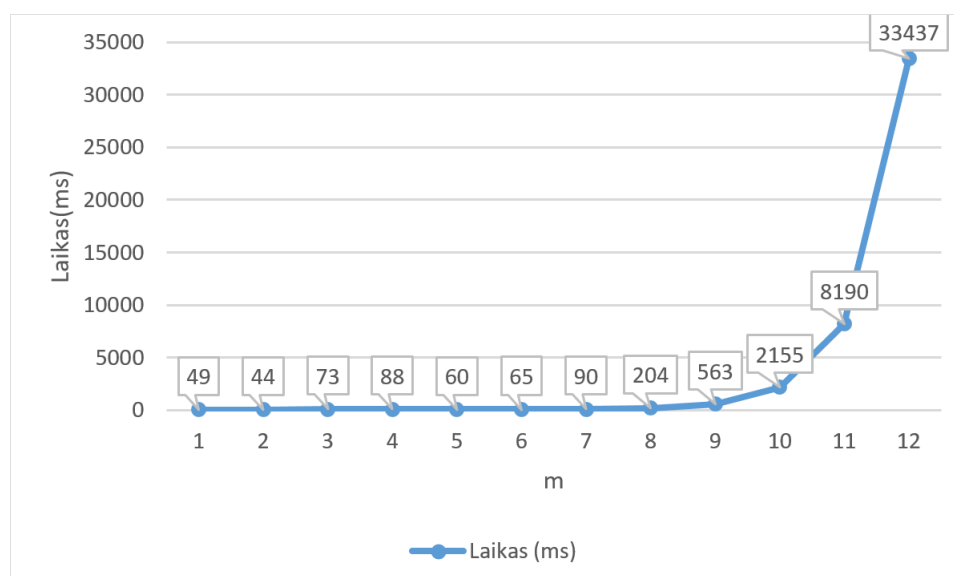
m	1	2	3	4	5	6	7	8
Klaidų, kurias padarius dekoduoamas vektorius tapo klaidingu, kiekis	1	1	2	4	8	16	32	64

Matant tokias tendencijas, gali būtų numanyti, kad tęsiant eksperimentą su  $m=9$ , rezultatas būtų  $2^7$ , su  $m=10$  -  $2^8$  ir t.t.

### 8.4. Teksto siuntimo laiko priklausomybė nuo m

Atliktas eksperimentas, kurio metu fiksuotas laikas siunčiant 157 simbolių ilgio tekstą:

*"Čia yra testinis tekstas kuriuo siekiama istestuoti ilgos tekstines žinutes kodavima bei dekodavima ir taip pat istestuoti kodavimo gebejima aptikti klaidas."*  
su skirtingais kodo parametrais m.



## 9. Naudota literatūra

<https://klevas.mif.vu.lt/~skersys/doc/ktkt/literatura13.pdf>

[https://en.wikipedia.org/wiki/BMP\\_file\\_format](https://en.wikipedia.org/wiki/BMP_file_format)