

1. Programos paleidimas

Programa yra paleidžiama **ReedMullerCode.exe** failu

2. Pradinių tekstų failai

Pradinių tekstų failai yra patalpinti “ReedMullerCode” direktorijoje

- Decoder aplankas
 - *Decoder.cs* – Rydo-Mulerio kodų dekodavimo įgyvendinimas
 - *DecoderExtensions.cs* – Dekoduotojo pagalbinių metodų klasė
- Encoder aplankas
 - *Encoder.cs* – Rydo-Mulerio kodų kodavimo įgyvendinimas
- Helpers aplankas
 - *Helpers.cs* – Pagalbinių metodų klasė, padedanti apdoroti naudotojo įvestus duomenis
 - *PictureHelpers.cs* – Pagalbinių metodų klasė, įgyvendinanti nuotraukos kodavimą, siuntimą kanalu bei dekodavimą. Taip pat neužkoduotos nuotraukos siuntimą kanalu
 - *TextHelpers.cs* – Pagalbinių metodų klasė, įgyvendinanti teksto kodavimą, siuntimą kanalu bei dekodavimą. Taip pat neužkoduoto teksto siuntimą kanalu
 - *VectorHelpers.cs* – Pagalbinių metodų klasė, įgyvendinanti vektoriaus kodavimą, siuntimą kanalu bei dekodavimą
- Matrices aplankas
 - *GeneratorMatrix.cs* – Generuojančios matricos klasė
 - *HadamardMatrix.cs* – Hadamardo matricos klasė
 - *Matrix.cs* – Matricos klasė
 - *MatrixExtensions.cs* – Pagalbinių matricos metodų klasė
 - *WeighingMatrix.cs* – Svertinės matricos klasė

3. Vartotojo sąsaja

Vartotojas gali pasirinkti, su kokio tipo duomenimis norės dirbti – vektoriumi, simbolių eilute ar paveikslėliu.

Darbas su vektoriais:

1. Į laukelį “m” įrašomas norimas m parametro dydis – tai skaičius nuo 1 iki 12 (su didesniais parametrais programa lūžta dėl atminties trūkumo)
2. Vartotojas įveda vektorių, sudarytą iš 0 ir 1. Jei vektorius sudarytas iš netinkamų simbolių arba vektoriaus ilgis neteisingas, programa vartotojui parodo atitinkamą pranešimą
3. Vartotojas spaudžia mygtuką „Encode“. Programa užkoduoja vektorių ir parodo jį ekrane
4. Vartotojas įveda klaidos tikimybę ir spaudžia mygtuką „Send“
5. Vartotojui sistema parodo iš kanalo išėjusį vektorių bei klaidų kiekį bei pozicijas
6. Vartotojas gali pakeisti iš kanalo išėjusį vektorių
7. Vartotojas spaudžia „Decode“ mygtuką. Sistema dekoduoja ir parodo dekodotą vektorių.

Darbas su simbolių eilute:

1. Į laukelį “m” įrašomas norimas m parametro dydis – tai skaičius nuo 1 iki 12 (su didesniais parametrais programa lūžta dėl atminties trūkumo)
2. Vartotojas įveda norimą tekstą (gali būti kelių eilučių) bei klaidos tikimybę ir spaudžia mygtuką „Send“
3. Vartotojui parodoma, koks tekstas gautas suskaidžius tekstą į vektorius, juos užkodavus, siuntus pro kanalą ir dekodavus
4. Vartotojui parodoma, koks tekstas gautas tekstą siuntus pro kanalą (be kodavimo)

Darbas su paveikslėliu:

1. Į laukelį “m” įrašomas norimas m parametro dydis – tai skaičius nuo 1 iki 12 (su didesniais parametrais programa lūžta dėl atminties trūkumo)
2. Vartotojas pasirenka norimą BMP formato paveikslėlį
3. Vartotojui parodomas jo pasirinktas paveikslėlis
4. Vartotojas įveda klaidos tikimybę ir spaudžia mygtuką „Send“

5. Vartotojui parodoma, koks paveikslėlis gautas suskaidžius paveikslėlį į vektorius, juos užkodavus, siuntus pro kanalą ir dekodavus
6. Vartotojui parodoma, koks paveikslėlis gautas paveikslėlį siuntus pro kanalą (be kodavimo)

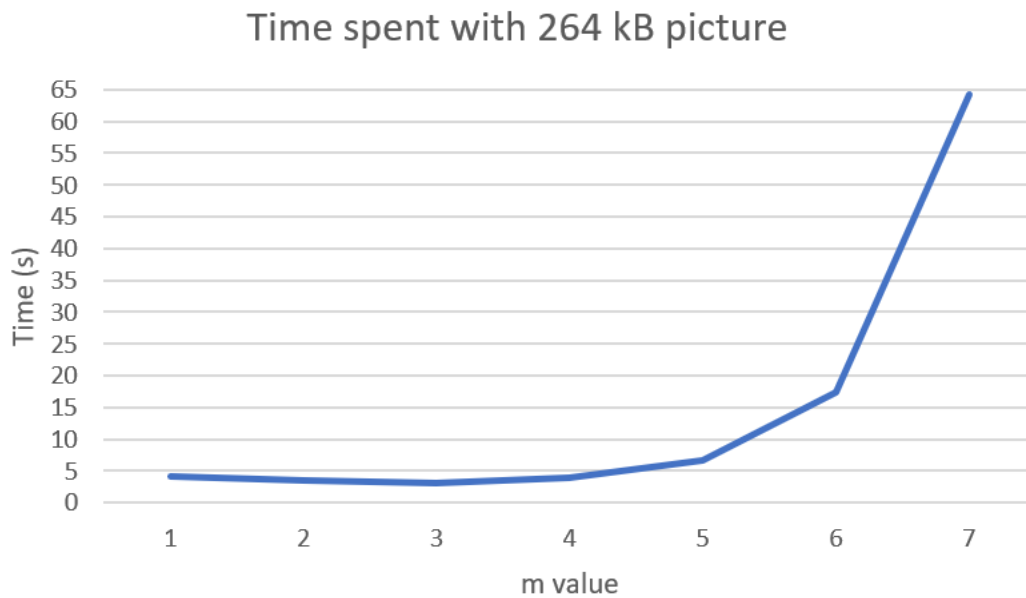
4. Programiniai sprendimai

- Tekstinės eilutės paverčiamos į bitų masyvą naudojant ASCII simbolių koduotę
- Bitai į tekstą paverčiami taip pat naudojant ASCII koduotę
- Jeigu bitus verčiant į vektorius, paskutiniam vektoriui prirašome 0
- Jei paskutinis vektorius turi prirašytą 0, jis nėra siunčiamas kanalu
- Vertimo į tekstą metu, prirašyti bitai nuimami
- Rydo-Mulero generuojančios ir Hadamardo matricos yra generuojamos tik jų prireikus, po to jos yra saugomos, susiejus jas su jas identifikuojančiais parametrais
- Hadamardo matricos yra generuojamos tik jų prireikus, po to jos yra saugomos, susiejus jas su jas identifikuojančiais parametrais
- Kroneckerio produktas yra gaunamas optimizuotu būdu – iš karto generuojant matricą, gautą iš matricos ir vienetinės matricos
- Dėl efektyvumo, vektorių kodavimas ir dekodavimas vyksta paraleliai
- Paveikslėlių kodavimas bei dekodavimas yra realizuotas tik su BMP tipo failais
- BMP failo headeriui naudojami 54 baitai. Jie kanalu nėra siunčiami
- Testams naudojama NUnit biblioteka

5. Bandymai

Pirmasis bandymas:

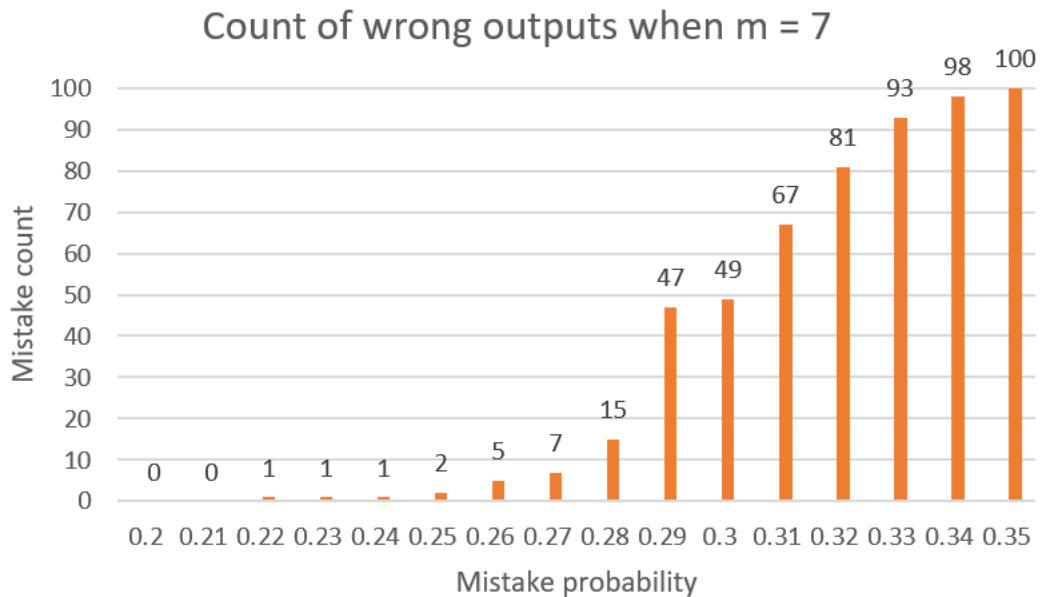
264 kB dydžio paveikslėlio kodavimo, siuntimo kanalu ir dekodavimo laiko priklausomybė nuo parametro m reikšmės. Klaidos tikimybė lygi 0.2



- Galime pastebėti, kad mažiausias laikas yra pasiekiamas, kai m reikšmė yra 3
- Kai parametro m reikšmė yra mažesnė nei 3, sistema turi aprodoti žymiai daugiau vektorių, dėl to laikas yra šiek tiek lėtesnis
- Kai parametro m reikšmė yra didesnė nei 3, sistema turi aprodoti didesnio dydžio matricas. Dėl to, kodavimo, siuntimo kanalu ir dekodavimo laikas sparčiai auga

Antrasis bandymas:

Tekstinė eilutė „abcdefghijklmnopqrstuvz1234567890“ (ilgis 32) yra užkoduojama, siunčiama kanalu su tam tikra klaidos tikimybe ir dekoduojama. Tikrinama, ar po visos veiksmų sekos, gauname tą pačią tekstinę eilutę (t.y. be jokių klaidų). Su kiekviena klaidos tikimybe buvo atlikta 100 bandymų. Parametro m reikšmė yra 7.



- Kai klaidos tikimybė yra mažesnė ar lygi 0.21, nepasitaikė nei vienos klaidos iš 100 bandymų
- Teksto iškraipymo tikimybė sparčiai kyla, kai klaidos tikimybė yra 0.29
- Nuo klaidos tikimybės 0.35 visos simbolių eilutės dekoduojamos klaidingai

6. Praleistas laikas

- Kodavimo aiškinimasis ir įgyvendinimas – 4h
- Dekodavimo aiškinimasis ir įgyvendinimas – 6h
- Vartotojo sąsajos kurimas – 3h
- Sistemos pritaikymas darbui su simbolių eilutėmis ir paveikslėliais – 3h
- Programos optimizavimas – 5h

7. Literatūros sąrašas

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