

Kodavimo teorija

Užduotis A15

Ataskaita

Darbą atliko:

Edas Lakavičius, 4 grupė

Užduoties realizacija

Programa pilnai realizuoja pirmąjį ir antrąjį scenarijus, yra užkoduojamas vartotojo pateiktas pranešimo vektorius, siunčiamas kanalu ir dekoduojamas. Trečiojo scenarijau pilnai įgyvendinti nepavyko, paveiksliukas yra koduojamas bei dekoduojamas, tačiau atvaizduoti tiek užkoduoto tiek dekoduito paveiksliuko nepavyko.

Bibliotekos

Jokios bibliotekos nebuvo naudojamos.

Programos paleidimas

Programa paleidžiama paleidus AK15.exe failą, esantį Coding/bin/Debug/ direktorijoje. Pilna direktorija Coding/bin/Debug/AK15.exe

Programos struktūra

Program.cs – programos įeities taškas, pagrindinė viską jungianti funkcija.

UiForm.cs- grafinio interfeiso klasė tuo pačiu apjungianti užkodavimo bei dekodavimo funkcionalumą.

OutputDecoder.cs- klasė, sukurianti objektą savyje saugantį dekoduoatą/dekoduojamą vektorių.

InitialData.cs- klasė, sukurianti objektą, sauganti 6 bitus informacijos (su pradinėm 0 reikšmėm), turinti pagalbinę funkciją galinčią grąžinti užkoduotą bitą.

InitialDataDecoder.cs- klasė, sukurianti objektą, sauganti 6 bitus informacijos (su pradinėm 0 reikšmėm), turinti pagalbinę funkciją galinčią grąžinti užkoduotą bitą, bei turinti mde apskaičiavimo pagalbinę funkciją.

Encoder.cs- užkodavimo funkcionalumą įgyvendinanti klasė.

Sender.cs- klasė, saugojanti funkcijas(bitų keičiamų pozicijų radimo ir bitų keitimo).

Vartotojo sąsaja

Vartotoja sąsają susideda iš 3 dalių:

1) Atitinka pirmąjį scenarijų

Vartotojas laukelyje įveda dvejetainį vektorių bei paspausdamas **Encode** mygtuką jį užkoduoja. Prieš siųsdamas užkoduotą vektorių kanalu vartotojas taip pat turi įvesti klaidos tikimybę. Paspaudžius **Send** mygtuką žinutė yra siunčiama kanalu. Parodomi klaidingai parašytų sąrašo bitų pozicijos. Paspaudžius **Decode** mygtuką kanale, esanti žinutė yra dekoduojama.

2) Atitinka antrąjį scenarijų

Veikia labai panašiai, kaip ir pirmąjį scenarijų įgyvendinanti dalis, tačiau koduojami, siunčiami ir dekoduojami vektoriai jau yra teksto simboliai. Taip pat prisideda 2 papildomi informacijai skirti rodyti laukai – skirti rodyti vektoriui, nenaudojant kodo.

3) Atitinka trečiąjį scenarijų

Kadangi šis scenarijus pilnai neveikia, tai galima iš savo kompiuterio pasirinkti paveiksluką – jį atvaizduoja specialus laukas. Taip pat kaip ir visuose scenarijuose vartotojas gali įvesti klaidos tikimybę, deja jokio rezultato vartotojas nesulaukia.

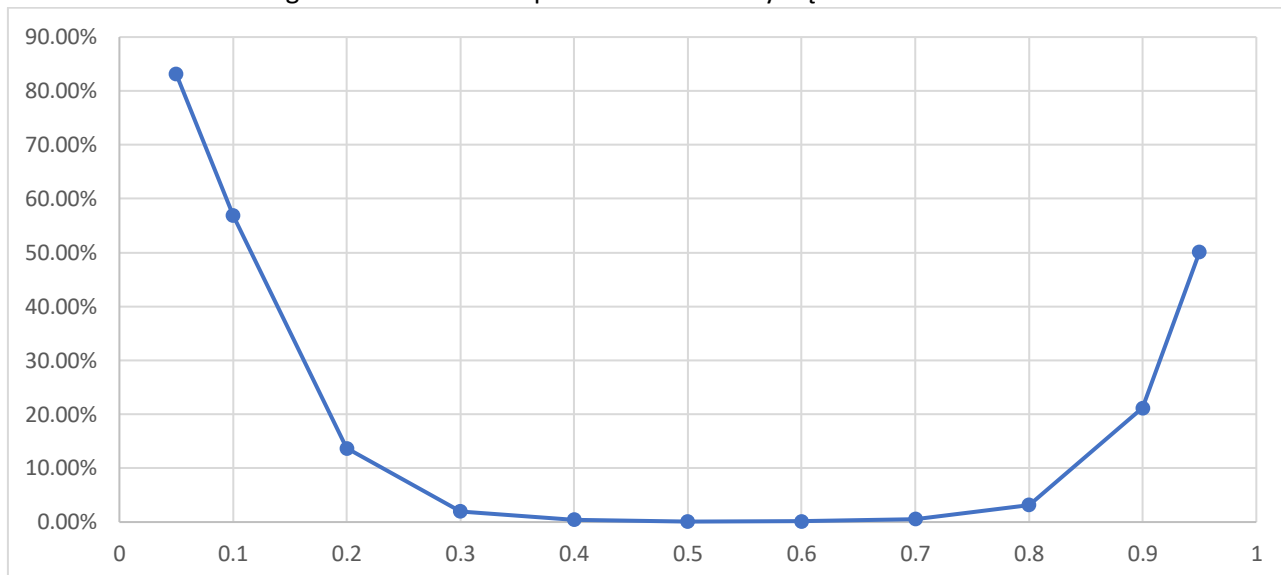
Programiniai sprendimai

- 1) Vartotojo įvesties nuskaitymas – vartotojo įvestas vektorius yra saugomas String tipo kintamajame, kuris vėliau naudojamas pasiversti į specialų skaičių masyvą, atitinkantį bitus.
- 2) Vektorių užkodavimas – kiekvienas vartotojo įvestas vektorius yra modifikuojamas, kaip minėjau yra paverčiamas į skaičių masyvą, atitinkantį bitus, tuo pačiu yra pridedami 6 papildomi bitai su 0 reikšme. Tada yra iteruojama per atnaujintą vartotojo vektorių, vykdant užkodavimo algoritmą. Kiekvienas atnaujintame vektoriuje esantis bitas yra paverčiamas į 2 bitus bei bito reikšmė įrašoma į pradinį 6 elementų sąrašą, pradedant nuo pradžios.
- 3) Vektorių siuntimas kanalu – kiekviena bitų formato skaičių masyve esanti reikšmė yra iškraipoma su klaidos tikimybe. Sugeneruojamas atsitiktinis double tipo skaičius iš [0,1] intervalo ir jeigu klaidos tikimybė yra didesnė už sugeneruotą skaičių, šio masyvo elemento pozicija yra išsaugoma, vėliau yra iteruojama per išsaugotas pozicijas ir atitinkamai iškraipomos reikšmės – jeigu elemento pozicija yra išsaugota jo reikšmė atitinkamai paverčiama priešinga tikrajai (jeigu reikšmė buvo 0, tai siunčiamas 1 ir atvirkščiai)
- 4) Vektorių dekodavimas – dekodavimo funkcijai yra paduodamas kanale, esantis vektoriaus variantas. Per šį sąrašą yra iteruojama. Taip pat yra naudojami 2 papildomi sąrašai – vienas bito reikšmei gauti, kitas MDE reikšmei gauti. Atitinkamai kiekvienos iteracijos metu yra redaguojami šie sąrašai – į pirmąjį tiesiog įtraukiamas sąrašo elementas, į antrąjį MDE reikšmė. Į rezultatą yra įtraukiamas 1 jeigu 1 skaičius yra nelygis arba 0 jeigu 1 nėra arba jų skaičius yra lyginis. Galų gale iš gauto rezultato pašalinam pirmus 6 elementus, rezultatas yra konvertuojamas į String tipą ir atvaizduojamas vartotojui.

Eksperimentai

Teisingai dekoduoja vektoriaus priklausomybė nuo klaidos tikimybės kt.

Pasirinktas vektoriaus ilgis – 12. Kiekvienam parametrui kt bandymų skaičius – 100.



Klaidos tikimybė artėjant 0.5 stipriai krenta pasiekimo procentas.

Išvada: prasminga naudoti tik esant labai mažai arba labai didelei iškraipymo galimybei.

Naudotos literatūros sąrašas

<https://klevas.mif.vu.lt/~skersys/20r/ktkt/KTKT.pdf>

<https://klevas.mif.vu.lt/~skersys/doc/ktkt/literatura23.pdf>