

Kodavimo teorijos programos ataskaita

Parengė Dominykas Jovaiša

Turinys

Realizuotos ir nerealizuotos užduoties dalys.....	3
Trečiųjų šalių bibliotekos	3
Laiko užtrukimas	3
Programos paleidimas	4
Programos tekstų failai	4
Vartotojo sąsajos aprašymas	4
Programiniai sprendimai.....	4
Eksperimentų aprašymas.....	5
Literatūra	6

Realizuotos ir nerealizuotos užduoties dalys

- Pirma, antra ir trečia dalys realizuotos.

Trečiųjų šalių bibliotekos

Paveiksluko rodymui:

- `import javax.imageio.ImageIO;`
- `import java.awt.image.BufferedImage;`

Programos „exceptionams“:

- `import java.io.*;`

Skaitymui iš konsolės:

- `import java.util.Scanner;`

Atsitiktinio skaičiaus generavimui:

- `import java.util.Random;`

Lengvesniam darbui su Masyvais:

- `import java.util.ArrayList;`

Paveiksluko dekodavimui:

- `import java.math.BigInteger;`

Laiko užtrukimas

Literatūros skaitymas, pratybų apie programą aiškinimasis – 4 valandos,

Programavimas - 20 valandų

Pirma dalis – 16 valandų

Antra dalis – 10 valandų

Trečia dalis – 12 valandų

Ataskaitos ruošimas – valanda

Bendrai: 63 valandos

Programos paleidimas

Visi reikalingi duomenis turi būti viename aplanke.

Programa pasileidžia paprastai, tereikia paleisti main.java klasę. Parametrų nėra, programa duomenis ima iš tekstinio failo.

Pirmos dalies tekstinis failas vadinasi: „1dalis.txt“. Pirmoje eilutėje reikia suvesti tokius duomenis: M, R, klaidos tikimybė(0-10000), informacija siunčiama kanalu.

Antros dalies tekstinis failas vadinasi: „2dalis.txt“. Pirmoje eilutėje reikia suvesti tokius duomenis: M, R, klaidos tikimybė(0-10000). Toliau kitose eilutėse įvedamas pranešimas(pranešimas gali būti ne tik iš vienos eilutės).

Trečios dalies tekstinis failas vadinasi: „3dalis.txt“. Pirmoje eilutėje reikia suvesti tokius duomenis: M, R, klaidos tikimybė(0-10000). Paveikslukas, su kuriuo dirbsime vadinasi „out.bmp“

Programos tekstų failai

Visi minėti failai „Programos paleidimas skiltyje“. Visa programa realizuota „main.java“ klasėje.

Visi failai turi būti „C:\Users\Vartotojas\Desktop\kodavimoLogika\src“ aplanke

Paveikslukas turi būti „C:\Users\Vartotojas\Desktop\kodavimoLogika\src\out.bmp“.

Vartotojo sąsajos aprašymas

Programos pradžioje, programa paprašo įvesti skaičių : „1“ , „2“ arba „3“. Atitinkamai, programa atliks pasirinktą dalį.

Visi pradiniai duomenys paduodami anksčiau minėtuose tekstiniuose failuose. 1 dalies atveju, kai programa prasiunčia vektorių kanalu, programa išspausdina vektorių gautą iš tunelio ir paklausia, ar vartotojas nori pakeisti kanalu gautą vektorių(su y/n logika). Jei vartotojas į konsolę įveda „y“, programa paprašo vartotojo įvesti, kaip jis nori, kad atrodytų iš kanalo gautas vektorius. Pvz.: programa turi „11111111“ užkoduotą vektorių, klaidos tikimybė yra, tarkim, 1-2500 t.y 25%, todėl programa po tunelio gavo „00111111“ vektorių. Programa jį išspausdina ir paklausia vartotojo, ar jis nori pakeisti šį vektorių. Vartotojui į konsolę įrašius „y“, programa paprašo įvesti vektorių, taigi vartotojas įveda „11001111“. Tuomet programa tęsia darbą su šiuo užkoduotu ir kanalu prasiųstu vektoriumi. Jei vartotojas įveda „n“, programa tiesiog darbą tęsia su iš kanalo gautu vektoriumi.

Programa baigus darbą duomenis išspausdina konsolėje.

Antroje ir trečioje dalyje tereikia pakeisti duomenis, minėtus „Programos paleidimas“ skiltyje.

Programiniai sprendimai

Nepilnas vektorius:

Jei 2 dalyje tekstą suskaidžius vektoriais, negaunamas pilnas vektorius, tuomet programa pabaigoje prirašo papildomų "0", kol vektorius tampa reikiamo ilgio. Poto programa šiuos papildomus nulius pašalina.

Skaidymas į vektorius:

Programa suka "loop" ciklą, iš "String" su duomenimis paimdama reikiamo ilgio vektorių. Su šiuo vektoriumi atlieka reikiamas operacijas ir jį deda į galutinio rezultato "String".

Vektorių siuntimas kanalu:

Vektorius paduodamas į "Tunnel" funkciją, kuri galimai pakeičia atskirai kiekvieną jo bitą, ir gražina galimai iškraipytą vektorių.

Eksperimentų aprašymas

1 Eksperimentas.

Tikriname pirmąją dalį. Į tekstinį dokumentą "1dalis" įrašome duomenis: 3 1 1000 1111

Paliedžiame program air pasirenkame "1", kad programa vykdytų pirmąją dalį.

Gauname užkoduotą vektorių 01101001. Ši vektorių siunčiame kanalu ir gauname 11101001.

Įrašome "n", nes nenorime keisti vektoriaus.

Matome, kad yra padaryta viena klaida. Programa vektorių dekoduoja ir gauna vektorių 1111. Programa teisingai ištaisė klaidas.

2 Eksperimentas.

Programos veikimo laikas.

Tikrinsime, kaip 2 dalies laikas priklauso nuo teksto kiekio.

Pasirenkame koduote RM(5,2).

Pirmiausia, į "2dalis" tekstini failą įrašome parametrus: 5 2 1000

Šie parametrai nekis eksperimento metu.

1 Bandymas. Įrašome tekstą(447 simboliai):

vykdome pirma bandymą, žiurim kiek užtruks laiko. vykdome pirma bandymą, žiurim kiek užtruks laiko.
vykdome pirma bandymą, žiurim kiek užtruks laiko.

vykdome pirma bandymą, žiurim kiek užtruks laiko. vykdome pirma bandymą, žiurim kiek užtruks laiko.
vykdome pirma bandymą, žiurim kiek užtruks laiko.

vykdome pirma bandymą, žiurim kiek užtruks laiko. vykdome pirma bandymą, žiurim kiek užtruks laiko.
vykdome pirma bandymą, žiurim kiek užtruks laiko.

(tekstas iš trijų eilučių).

Program užtruko 289ms

2 bandymas Įrašome 5 kartus ilgesnį tokį pat tekstą(2235 simboliai).

Programa užtruko 780ms

2 bandymas Įrašome 10 kartų ilgesnį tokį pat tekstą kaip pirmame bandyme(4470 simbolių).

Programa užtruko 1612ms

Išvada

2 dalies užduoties su parametrais: 5 2 1000, grafikas:



Literatūra

- http://www.mif.vu.lt/lmd/kodai_sifrai.pdf
- Pratybų, apie A7 kodavimą Įrašas