

A PROGRAMOZÁS ALAPJAI 1
(BMEVIEEAA00, 2024/25/1)
NAGYHÁZI FELADAT

Shanon-Fano kódoló és dekódoló program

készítette:
Ferencz Péter
(RFG7SN)



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Mérnökinformatikus Bsc
2024 Október

Tartalomjegyzék

1. Specifikáció	2
1.1. A program célja	2
1.2. Felhasználói interakció	2
1.2.1. kódolás (kódol)	2
1.2.2. dekódolás (dekodol)	2
1.3. A program által elfogadott kapcsolók-ek	3
1.3.1. Bemenet	3
1.3.2. Kimenet	3
1.3.3. Kódtábla	3
1.3.4. Statisztika	3
1.4. A program kimenete	4

1. fejezet

Specifikáció

1.1. A program célja

A program célja tetszőleges adat tömörítése majd ezek kitömörítése információvesztés nélkül. Ennek megvalósítására a Shanon-Fano tömörítő algoritmust ¹ ² alkalmazza.

1.2. Felhasználói interakció

A felhasználó két üzemmódot választhat ki a program futtatásakor: kódolás vagy dekódolás. Ezeket az első parancssori argumentumban a 'kodol' és 'dekodol' kulcsszavakkal tudja kiválasztani.

1.2.1. kódolás (kodol)

Kódoló üzemmódban a bemenetet (lásd [Bemenet](#)) a Shanon-Fano kódoló algoritmust alkalmazva írja a kimenetre (lásd [Kimenet](#)) a kódolt adatot.

```
program kodol --bemenet <fájl> --kimenet <fájl> --kodtabla <fájl>
```

1.2.2. dekódolás (dekodol)

Dekódoló üzemmódban a bemenetet (lásd [Bemenet](#)) a Shanon-Fano dekódoló algoritmust alkalmazva írja a kimenetre (lásd [Kimenet](#)) a dekódolt adatot.

```
program dekodol --bemenet <fájl> --kodtabla <fájl> --kimenet <fájl>
```

¹C. E. Shannon, „A Mathematical Theory of Communication”, 1948

²Robert M. Fano, „The Transmittion of Information”, 1949

1.3. A program által elfogadott kapcsolók-ek

A program futása során tetszőleges futtatást befolyásoló kapcsolókat (flageket) beállíthatunk. Ezek sorrendje tetszőlegesen választható.

1.3.1. Bemenet

Parancssori megnevezés: `--bemenet <forrásfájl>`

Opcionális paraméter. Ha nincs megadva, de a program egy figyelmeztető üzenet kíséretében folytatja a lefutást.

A fájl méretétől és tartalmától független a program lefutása.

Az azt követő paraméter megadja a forrásfájl elérési útvonalát. Ha nincs megadva, stdin-ról kér be új sorral lezárt szöveget.

1.3.2. Kimenet

Parancssori megnevezés: `--kimenet <célfájl>`

Opcionális paraméter. Ha nincs megadva, de a program egy figyelmeztető üzenet kíséretében folytatja a lefutást.

A fájl méretétől és tartalmától független a program lefutása.

Az azt követő paraméter megadja a célfájl elérési útvonalát. Ha nincs megadva, stdout-ra írja ki a program kimenetét.

1.3.3. Kódtábla

Parancssori megnevezés: `--kodtabela <fájl>`

Azt határozza meg, hogy a program kiírja-e a kódtáblát. A program alapjáraton nem írja ezt ki, ha a kapcsoló nincs beállítva.

A működést üzemmódtól függően változtatja:

[kódolás \(kodol\)](#) esetén a [Kimenet](#) előállítására használt kódtáblát kiírja egy általunk meghatározott fájlba.

[dekódolás \(dekodol\)](#) esetén a [Bemenet](#) egy előre meghatározott kódtáblát tartalmazó fájl alapján dekódolja.

A kódtábla fájl struktúrája:

Egy karakter, melyet egy szóköz válasz el egy tetszőleges hosszúságú bináris számsorozattól. Tehát minden sor első karaktere az azt szóközzel elválasztott kóddal van kódolva. Például:

a 011

b 001

1.3.4. Statisztika

Parancssori megnevezés: `--statisztika`

Opcionális paraméter.

Azt határozza meg, hogy a program kiírjon-e további számításokat a program hatékonyságára vonatkozólag.

Az alábbi számítások történnek kiírásra:

- Tömörítés mértéke: bemenet mérete a tömörített adat méretéhez képest

- Kódtábal mérete: Elgymástól eltérő kódok száma
- Kódok mérete: legrövidebb kód, leghosszabb kód, kódok átlagos mérete
- Fa mérete: A generált fa mérete

1.4. A program kimenete

Sikeres futtatás esetén a program a [A program által elfogadott kapcsolók-ek](#) pontban meghatározott viselkedés szerint működik. Sikertelen futtatás esetén a konzolra kiíródik a probléma és egy nem nullás kilépési kóddal a program megáll.