GZIP KOMPRESIJA

Jelena Dokić RA98-2015

Kompresija podataka

- Šifrovanje podataka korišćem manje bita nego što je korišćeno u originalnom formatu
- Memorija koju zauzimaju podaci se smanjuje
- Može biti sa gubicima podataka i bez gubitaka
- U slučaju gzip kompresije podaci se ne gube

Originalna gzip kompresija

- GZIP je file format i aplikacija koja se koristi za kompresiju i dekompresiju
- Najbolje rezultate pokazuje u kompresiji tekstualnih datoteka i .tar datoteka (3 do 4 puta manja velicina), a može se primenjivati i na ostalim vrstama datoteka
- Kompresija zavisi od količine redundantnosti u datotekama i veličine datoteke
- Bazira se na DEFLATE algoritmu, koji je kombinacija LZ77 ili LZ78 algoritma i Huffman coding algoritma
- Najčešće se javlja u obliku konzolne aplikacije

Originalna gzip kompresija

- Prva verzija objavljena je 31. oktobra 1992. godine
- Tvorci: Jean-loup Gailly i Mark Adler
- Nastala je kao Open-Source besplatna zamena za prethodno korišćene algoritme za kompresiju i namenjena je za Unix operativne sisteme
- g = GNU u nazivu kompresije
- Trenutna verzija: 1.9 objavljena je 7. januara 2018. godine
- Napisana je u c programskom jeziku
- Pored GNU verzije gzip kompresije javlja se i OpenBSD verzija gde g = gratis
- Na osnovu gzip kompresije nastali su: HTTP kompresija, PNG kompresija, 7 zip i AdvanceComp

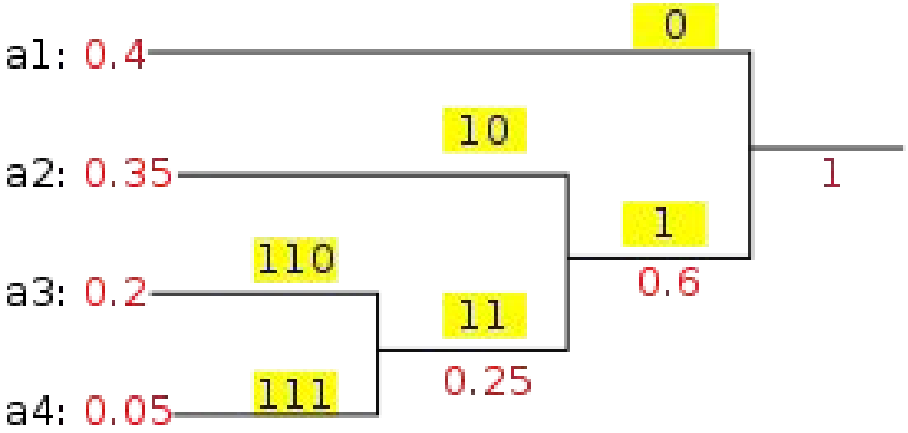
LZ78 algoritam

- pronalazi karaktere ili niz karaktera koji su se već pre nalazili u ulaznom fajlu i menja ih referencom na njihovu prvu pojavu. Referenca na prvu pojavu se sastoji od dva broja: koda, koji označava niz karaktera koji se pre toga pojavio i dužine, koja prikazuje koliko znakova iz prve pojave se nalazi u trenutnoj pojavi
- Blah blah blah blah blah! - ulazni string (uočavamo delove koji se ponavljaju)
- Blah blah b - prvo ponavljanje dela lah b
- Prethodne pojave koje se referenciraju se čuvaju u rečniku radi lakšeg indeksiranja : dictionary[...] = {code, character}

Huffman code algoritam

- Svakom slogu iz LZ78 rečnika se dodeljuje binarni kod
- Kodovi su različite dužine, srazmerno učestalosti pojavljivanja slogova
- a1,a2,a3,a4 - slogovi iz LZ78 rečnika

Zbog najučestalijeg pojavljivanja a1 sloga,on dobija najkraci kod,dok se slogovi a3 i a4 pojavljuju redje pa zato dobijaju znatno duže kodove. Na osnovu ukupnog pojavljivanja i same dužine pokazuje se da ovaj način kodiranja smanjuje dužinu zapisivanja celog ulaznog fajla, u odnosu na zapisivanje gde svaki slog ima kod jednake dužine:



- jednaki kodovi: 4 * 2 = 8
- Huffman code: 4 * (0.4 * 1 + 0.35 * 2 + (0.2 + 0.05) * 3) = 7.4

Dekompresija

- Obrnut postupak od kompresije
- Vraća datoteku u prvobitno stanje
- Zbog jedinstvene identifikacije slogova u oba algoritma, dekompresija se obavlja pronalaženjem slogova i vraćanjem u prvobitno stanje pomoću rečnika

Moja implementacija

- Rađena je u programskom jeziku Python 3.6.4
- Koristi LZ78 algoritam i Huffman code algoritam
- Kompresuje tekstualne datoteke, .tar datoteke i direktorijume tako što rekurzivno prolazi kroz njih i kompresuje svaku datoteku posebno (tako radi i originalni gzip algoritam)
- Omogućava davanje imena izlaznoj datoteci i čuva i prvobitnu i kompresovanu verziju (što originalni gzip ne omogućava)

Moja implementacija

Pre pokretanja:

- export PATH=$PATH":."
- #!/usr/bin/env python
- Ove dve komadne omogucavaju pokretanje programa bez rezervisane reči python (način za pokretanje programa pisanih u python-u) i “./<program>” (za pokretanje izvršnih aplikacija)

Moja implementacija

- Pokretanje:
  - compress [opcija] <ulazna_putanja> <izlazna_putanja>
  - decompress <ulazna_putanja> <izlazna_putanja>
- Opcija “-d” se stavlja pri kompresiji direktorijuma dok se za kompresiju datoteka izostavlja
- Za ulaznu i izlaznu putanju se navodi relativna adresa direktorijuma ili datoteka

Compress koraci:

```
def compress(inFilename,zipFilename):
    emptyFile(zipFilename)
    d = readFromFile(inFilename)              #preuzima sadrzaj fajla
    dictionary = makeLZ78Dictionary(d)        #vraca listu slogova
    dictionary = makeDictionary(dictionary) #pravi dictionary od slogova
    dictionary = countWords(d,dictionary)   #broji koliko ima ponavljanja
    sortDictionary(dictionary)                #sortira na osnovu broja ponavljanja
    makeHuffmanCodes(dictionary)              #pravi huffman stablo za sortiran dictionary
    s = LZ78compress(d,dictionary)            #kompresovan kod
    writeDictionaryToFile(zipFilename,dictionary)       #upisivanje recnika u file
    writeCodeToFile(zipFilename,s)            #upisivanje koda u file
    checkSize(inFilename, zipFilename)        #proverava uspesnost kompresije
```

Decompress koraci:

```
def decompress(zipFilename,unzipFilename):
    emptyFile(unzipFilename)
    d = readContext(zipFilename)                          #ucitavanje fajla
    dictionary = readDictionary(zipFilename)              #ucitava dictionary
    dictionary = decompressLZ78dictionary(dictionary)     #pravljenje recnika(kao strukture)
    s = decompressLZ78Codes(d,dictionary,dictionary[len(dictionary)-1].binCode,unzipFilename)
            #dekompresija glavnog dela
    writeCodeToFile(unzipFilename,s)              #ispis glavnog dela u file
```

Primer compress:

```
$ ls -lh
drwxr-xr-x 4 jelena jelena 4.0K Feb  6 17:38 test/
-rw-r--r-- 1 jelena jelena   92 Jan 30 16:00
codeDecompress.py
-rw-r--r-- 1 jelena jelena  121 Jan 30 16:00 code.py
-rwxr-xr-x 1 jelena jelena 1.8K Feb  6 18:32 compress*
-rwxr-xr-x 1 jelena jelena  640 Feb  6 18:55 decompress*
-rw-r--r-- 1 jelena jelena 9.9K Feb  6 17:13 functions.py
-rw-r--r-- 1 jelena jelena  35K Jan 27 14:48 LICENSE
-rw-r--r-- 1 jelena jelena  424 Jan 30 16:03 README.txt
-rw-r--r-- 1 jelena jelena 8.8K Feb  6 17:44 tartest.gzip
```

```
$ compress test/index.html index.gzip
```

```
$ ls -lh
drwxr-xr-x 4 jelena jelena 4.0K Feb  6 17:38 test/
-rw-r--r-- 1 jelena jelena   92 Jan 30 16:00
codeDecompress.py
-rw-r--r-- 1 jelena jelena  121 Jan 30 16:00 code.py
-rwxr-xr-x 1 jelena jelena 1.8K Feb  6 18:32 compress*
-rwxr-xr-x 1 jelena jelena  640 Feb  6 18:55 decompress*
-rw-r--r-- 1 jelena jelena 9.9K Feb  6 17:13 functions.py
-rw-r--r-- 1 jelena jelena 2.9K Feb  6 19:25 index.gzip
-rw-r--r-- 1 jelena jelena  35K Jan 27 14:48 LICENSE
-rw-r--r-- 1 jelena jelena  424 Jan 30 16:03 README.txt
-rw-r--r-- 1 jelena jelena 8.8K Feb  6 17:44 tartest.gzip
```

```
$/tests ls -lh
drwxr-xr-x 3 jelena jelena 4.0K Feb  6 17:13
foldertest/
drwxr-xr-x 4 jelena jelena 4.0K Feb  6 17:13 tartest/
-rw-r--r-- 1 jelena jelena 3.5K Jan 30 16:02 index.html
-rw-r--r-- 1 jelena jelena   18 Jan 30 16:02 proba.txt
-rw-r--r-- 1 jelena jelena 1.4K Jan 30 16:02 song
-rw-r--r-- 1 jelena jelena  10K Feb  6 17:38
tartest.tgz
```

Primer decompress:

```
$ ls -lh
drwxr-xr-x 4 jelena jelena 4.0K Feb  6 17:38 test/
-rw-r--r-- 1 jelena jelena   92 Jan 30 16:00
codeDecompress.py
-rw-r--r-- 1 jelena jelena  121 Jan 30 16:00 code.py
-rwxr-xr-x 1 jelena jelena 1.8K Feb  6 18:32 compress*
-rwxr-xr-x 1 jelena jelena  640 Feb  6 18:55 decompress*
-rw-r--r-- 1 jelena jelena 9.9K Feb  6 17:13 functions.py
-rw-r--r-- 1 jelena jelena 2.9K Feb  6 19:25 index.gzip
-rw-r--r-- 1 jelena jelena  35K Jan 27 14:48 LICENSE
-rw-r--r-- 1 jelena jelena  424 Jan 30 16:03 README.txt
-rw-r--r-- 1 jelena jelena 8.8K Feb  6 17:44 tartest.gzip
```

```
$ decompress index.gzip index1.html
```

```
$ ls -lh
drwxr-xr-x 4 jelena jelena 4.0K Feb  6 17:38 test/
-rw-r--r-- 1 jelena jelena   92 Jan 30 16:00
codeDecompress.py
-rw-r--r-- 1 jelena jelena  121 Jan 30 16:00 code.py
-rwxr-xr-x 1 jelena jelena 1.8K Feb  6 18:32 compress*
-rwxr-xr-x 1 jelena jelena  640 Feb  6 18:55 decompress*
-rw-r--r-- 1 jelena jelena 9.9K Feb  6 17:13 functions.py
-rw-r--r-- 1 jelena jelena 3.5K Feb  6 19:32 index1.html
-rw-r--r-- 1 jelena jelena 2.9K Feb  6 19:25 index.gzip
-rw-r--r-- 1 jelena jelena  35K Jan 27 14:48 LICENSE
-rw-r--r-- 1 jelena jelena  424 Jan 30 16:03 README.txt
-rw-r--r-- 1 jelena jelena 8.8K Feb  6 17:44 tartest.gzip
```

```
$ diff test/index.html index1.html
$
```

```
// ako komanda diff ne
ispiše nista,to znači da su
ulazne datoteke jednake
```

Reference:

- https://git.savannah.gnu.org/cgit/gzip.git/ - gzip izvorni kod
- http://www.zlib.net/feldspar.html - deflate algoritam
- http://www.infinitepartitions.com/art001.html - gzip format
- https://en.wikipedia.org/wiki/LZ77_and_LZ78 - LZ77 i LZ78
- https://en.wikipedia.org/wiki/Huffman_coding - Huffman coding

KRAJ

Hvala na pažnji!