NTNU

Kunnskap for en bedre verden

Øving 8

Algoritmer og Datastrukturer

Henrik Halvorsen Kvamme

11. november 2023

Innhold

1 Introduksjon

Denne rapporten beskriver utviklingen av et komprimerings- og dekomprimeringsprogram. Komprimering av data er en fundamental operasjon i informasjonsteknologi, som muliggjør effektiv lagring og overføring av data. I denne øvingen anvendes den velkjente komprimeringsteknikken Lempel-Ziv-Welsh (LZW), en videreutvikling av Lempel-Ziv (LZ). Formålet er å implementere disse algoritmene for å redusere størrelsen på filer uten å miste informasjon og deretter nøyaktig gjenopprette originaldataene via dekomprimering. Rapporten vil dekke teoretisk grunnlag, implementasjonsdetaljer, resultatene av komprimeringstester, og diskusjon av programmets effektivitet og anvendelighet.

2 Teori

2.1 Generell komprimeringsteori

Komprimering av filer er prosessen med å redusere filstørrelsen ved å anvende algoritmer som identifiserer og eliminerer redundans i dataene. To hovedtyper av komprimering er **lossless** og **lossy** komprimering. Mens *lossless* komprimering tillater en perfekt gjenopprettelse av originaldata, tillater *lossy* komprimering at noe data går tapt for å oppnå høyere komprimeringsrater. I denne oppgaven fokuserer vi på *lossless* komprimeringsmetoder.

2.2 Lempel-Ziv-Welsh (LZW)

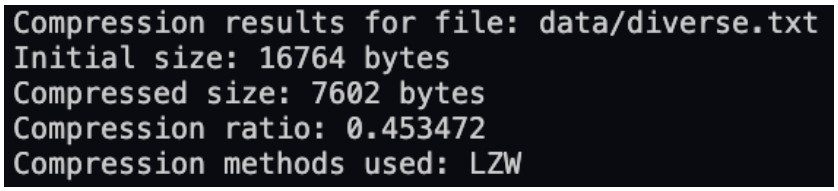
LZW er en komprimeringsalgoritme som reduserer filstørrelse ved å erstatte gjentakende data med kortere referanser. Den er populær i filformater som GIF og PDF.

2.3 Utfordinger med filkomprimering

Komprimeringsteknikker står overfor flere utfordringer, spesielt når det gjelder å håndtere forskjellige typer data. For eksempel, mens tekstfiler ofte komprimeres effektivt, kan binære filer eller allerede komprimerte filer (som JPEG-bilder) ikke komprimeres like mye og noen ganger kan de øke i størrelse når man prøver å komprimere dem. Derfor er det viktig å forstå datakonteksten og velge riktig komprimeringsalgoritme tilsvarende.

3 Resultater

Etter å ha implementert LZW komprimeringsalgoritmen, ble det utført flere tester for å evaluere effektiviteten. Her er et resultat på en av testfilene i oppgaven, 'diverse.txt':



Figur 1: Bare 45% av stlørrelsen!

4 Konklusjon

Jeg valgte å gjøre som de to tidligere oppgavene med å implementere en klasse for å lese in data fra fil. Den tar inn filnavn og bruker fstream for å lese og skrive data. Klassen har metode for å komprimere og dekomprimere filer.

Kildekoden ligger vedlagt i main.cpp. Et nytt objekt for den komprimerte filen blir oprettet for å vise at den ikke overfører tilstand fra da den komprimerte.x