

Sıkıştırma Algoritmaları Projesi

Kocaeli Üniversitesi
Bilgisayar Mühendisliği Bölümü

Sefa ÖZTÜRK – Faruk ARIĞ
180202036 - 180202009

ÖZET: Bu projede hedefimiz verilen dosyanın içeriğini istenilen sıkıştırma algoritmalarını kullanarak sıkıştırmak ve sıkıştırılan verileri kıyaslayarak hangi algoritmanın daha başarılı olduğu sonucuna ulaşmaktır. Projemizi geliştirirken LZ77 ve Deflate Sıkıştırma Algoritmaları kullanılmıştır.

Anahtar Kelimeler – Struct, Fonksiyon, Huffman, LZ77, Deflate

I. GİRİŞ

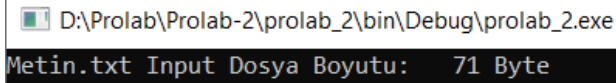
Programımızda bir klasör içinde bulunan `metin.txt`'yi okurken bir stringe almış oluyoruz. Bu dosyayı boyunca gezerek LZ77 ve Deflate algoritmalarını kullanıyoruz. Burda gerekli işlemleri yaptıktan sonra hangi sıkıştırma algoritması ne kadar sıkıştırma yapmış bunu ekrana bastırır ve output `.txt`'leri proje dosyası içerisine yazdırır.

II. YÖNTEM

Programımızı C programlama dili ile geliştirdik. Kodumuzda char yapısı, for-while döngüleri, if-else koşul durumları, struct yapısı ve bazı özel fonksiyonlar kullanarak programımızı geliştirdik.

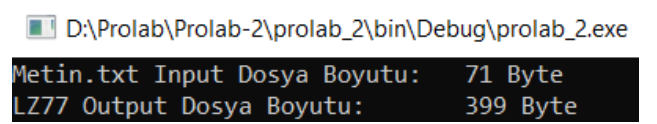
III. DENEYSEL SONUÇLAR

İlk başta kullanıcının oluşturduğu bir `metin.txt` dosyasını açıyoruz ve stringe atamış oluyoruz. Fseek ile dosyanın boyutunu hesaplarız. Ardından şekil.1 de ki gibi bir gerekli kısım konsolun bir kısmında çıkmaktadır.



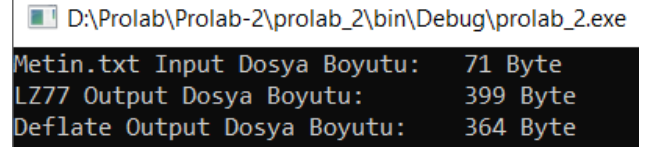
Şekil.1 Kullanıcının karşısına çıkan pencere.

LZ77 algoritmasını yaparken bu dosyayı boyunca gezmektedir. Bu sayede dosyanın sıkıştırılmış boyutunu hesaplar



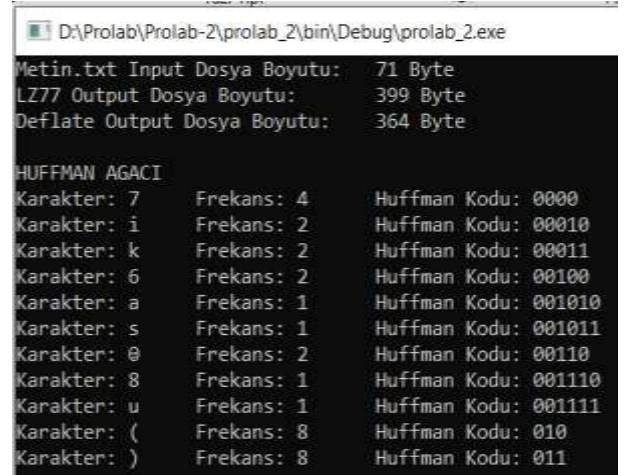
Şekil.1.1 LZ77 algoritmasının sonucu.

Aynı şekilde Deflate için gezerken aynı zamanda dosya boyutunu da hesaplamaktadır.



Şekil.1.2 Deflate algoritmasının sonucu.

Diğer bir Deflate için gerekli olan algoritma ise Huffman algoritmasıdır. Bunun da çıktısında karakterler, frekanslar ve Huffman kodları bulunmaktadır



Karakter	Frekans	Huffman Kodu
7	4	0000
i	2	00010
k	2	00011
6	2	00100
a	1	001010
s	1	001011
0	2	00110
8	1	001110
u	1	001111
(8	010
)	8	011

Şekil.1.3 Huffman yapısı.

Kullanılan algoritmalar sonucu oluşturulan output.txtler bastırılmıştır. Çıktıda da görüldüğü üzere LZ77 algoritmasındaki köşeli parantezler(token) içerisinde ayrılmış olan parametreler; İlk parametre kaç karakter geriye gideceğini, ikinci parametre benzerliğin uzunluğunu, üçüncü parametre benzerlikten sonra gelen karakteri göstermektedir.

```
*lz77_output - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
[0,0,C(P)][0,0,C(1)][0,0,C(m)][0,0,C(0)][0,0,C(u)][0,0,C( )]
[0,0,C(k)][0,0,C(A)][0,0,C(9)][0,0,C(Ä)][0,0,C(Y)][0,0,C(e)]
[0,0,C( )][0,0,C(y)][0,0,C(a)][0,0,C(z)][11,7,C(s)][0,0,C(i)]
[0,0,C(,)][0,0,C( )][0,0,C(b)][24,9,C(k)][0,0,C(A)][0,0,C(±)]
[0,0,C(Ä)][0,0,C(Y)][26,11,C(o)][0,0,C(r)][0,0,C(t)][0,0,C(a)]
[0,0,C(d)][0,0,C(a)][0,0,C( )][0,0,C(s)][0,0,C(u)][0,0,C( )]
[0,0,C(Ä)][23,12,C(k)][70,65,C( )]
```

Şekil.2 LZ77 .algoritmasının çıktısı.

Bir diğer çıktı ise Deflate çıktısıdır. Bunun oluşturulması içinde gerekli olan iki algoritma vardır ve bunlar Lzss ve huffmandır. Lzss, lz77 gibi benzer bir şekilde çalışmaktadır ancak sprint şeklinde biz bu çıktıları direk yazdırmayıp tutuyoruz ve sonunda ekler, sonra Huffman ile Deflate adı altında çıktılıyor. Huffman a göndermek içinde Lzssoutput stringinde bunları tutuyoruz.

```
*deflate_output - Not Defteri
Dosya Düzen Biçim Görünüm Yardım
001101000000110011010000001100000110001110011
1001010010100101001010010000010100110011
111000100011101111110111010101000110010
110001010110011001010111110001011010000
000111001010010100101001010010111110000
100101110111001111111001101101110110110
1101100010111101010010010111100100011111
101011101010111100000110111111101101001
011100110100111111010011010111000101011
```

Şekil.3 DEFLATE.algoritmasının çıktısı

IV. ALGORİTMALAR

1. LZ77 ALGORİTMASI

Bu dosyayı boyunca iki kez geziyor. İkincisinde diğer harflerle karşılaştırıyor. Eğer ki bu iki harf birbirine eşitse ve bir sonraki harf de birbirine eşitse kontrolün içine giriyor. Buradaki amaç eğer ki sadece bir harf benziyorsa token oluşturmasın, aslında token oluşuyor ancak benzerlik olmasın, yani uzunluğu 1 olan bir benzerliğimiz olmuyor. Ondan sonra bu kontrolün içine girdiğinde Tokenlar oluşturuyor. köşeli Parantezle birlikte. Bu tokenların mantığı köşeli parantezin içinde 3 tane

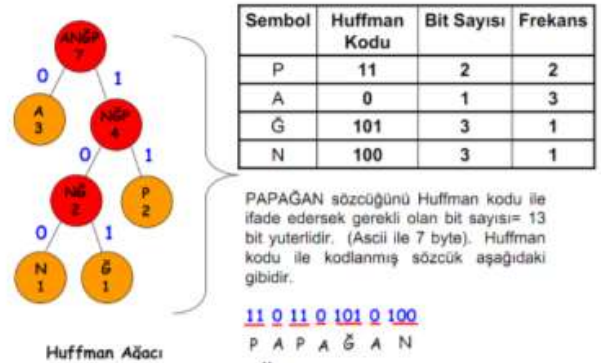
parametre var. İlk parametre kaç karakter geriye gideceği ikinci parametre.benzerliğin uzunluğu 3. parametre İse benzerlikten sonraki gelen karakter. Bunun sonuncunda da dosya uzunluğu ve output oluşuyor. Eğerki hiçbir yerde benzerlik yoksa direk sıfıra sıfır karakter şeklinde yazdırır.

2. DEFLATE ALGORİTMASI

Lzss kısmı çok benzer ancak sadece int değerlerin hesaplaması farklı ancak en önemli kısmı sprint kısmında bunları saklar ve birbirin sonuna ekler bunları da string şeklinde huffmana gönderir.

-HUFFMAN ALGORİTMASI-

Huffmanda bir tane dizi oluşturduk. Bu structın yapısında. Öncelikle bütün harfleri okuyup bu harfleri frekans Dizime atıyorum. Bu frekans Dizime atarken de bu dizinin keyleri karakterlerin ascii kodu olmuş oluyor. Değerleri kaç defa tekrar ettiyse frekansı olmuş oluyor bu frekansları dönerken dizimi oluşturuyorum artık bu frekansları mı aldım struct dizimi oluşturacağım burada da eğer ki daha önce hiç oluşturmadığı ise A eşit sıfıra ilk dizinin ilk elemanını oluşturuyorum yoksa Tmp diye struct alıp bunu insert de gönderiyorum insert de struct dizinin sonunda ekliyor. Bu Struct dizisini iki farklı şekilde kullanıyoruz. Önce bir dikey dizi olarak Next prev olarak kullanıyoruz. Tamamen harfleri ve frekanslarını sıralı şekilde yaptık. Ondan sonra sort fonk gönderdirildiğinde bu dikey diziyeye frekanslarına göre büyükten küçüğe sıralıyor sonra bir döngü çalışıyor. Bu dikey dizimizi oluşturduk şimdi bütün karakterleri frekansları ile birlikte bir dikey dizi yaptık bunu büyükten küçüğe doğru sıraladık şimdiki bu dizi sonuna kadar gidiyor. Buna göre de sağ ve sol dallarına ayırarak huffman ağacını oluşturmuş oluyoruz.



Bu algoritmanın oluşmasını sağlayan en büyük fonksiyonlardan biri de getCode fonksiyonu.

Getcode fonksiyonun parametrelerine baktığımızda buradan ilki ağacımızın ilk elemanı left Right elemanını tutuyor bufferbir tane String Lzssdeki alttaki Lzss

outputundaki Harf harf dönüyor Her harfini geccode da gönderiyor oradaki Huffman code karşılığını alıyor. Bu code karşılığını Getcode değerine eşitliyor Biz de onu yazduruyoruz Getcodeda yine bir tane gez tutuyoruz Bu geze Eğer ki sol ve sağ dallar boşsa en alta gelmişsin. Bu bir karakteri temsil ediyor demek. Bu karakter temsil ettiği karakterdir. Eğer ki parametrede C ise bunu kodunu eşitliyoruz stremp ile kodu prefix yapıyorum. Prefix de bunun Huffman kodunu buhmann kodu.

Prefix kodu nasıl oluşur? İlk baştakinde sol varsa getcode ile soluna gönderiyorum yani buradaki rekürsif fonksiyon bende solu var sola gönderiyorsa varsa sana gönderiyor sola gönderirken başına 0 sağa gönderirken başına 1 ekliyor. Bu şekilde yani biz istediğimiz bir karakteri huffman kodunu alabiliyoruz. Bunu da lzssout sıraylaher karakterin Huffman kodu ile değiştirerek yazdırıyoruz. Burada yine de deflatelemiş oluyoruz

V. SONUÇ

Bir dosyadan txt uzantılı dosya okumak ve bunun içerisinde işlemler yapabilmek. Bu dosya içerisindeki cümle ya da kelimeleri LZ77 ve DEFLATE algoritmaları kullanarak bunları anlamak ve bunları uygulamak. Algoritmalar arasında hangi algoritma daha hızlı ve verimli onu anlayıp uygulamaya dökmek. LZ77 ve LZSS arasındaki farkları öğrenmek.

VI. PROJE ESNASINDA KARŞILAŞILAN SORUNLAR

- Her ne kadar Huffman algoritmasını Veri yapıları dersinde teorik olarak görmüş olsak da proje geliştirirken Pratik olarak nasıl dönüştüreceğimizde zorlandık.
- LZ77 algoritması ve Deflate Algoritması hakkında kaynak bulmakta oldukça zorlandık, aynı zamanda algortimaları ilk defa gördüğümüz için uzun süre araştırma yaptık.
- Deflate algoritmasını geliştirirken belli kaynaklarda LZ77+Huffman belli kaynaklarda LZSS+Huffman olduğunu gördük . Bu da kafa karışıklığına neden oldu.

VI. KAYNAKÇA

- [1] Bilgisayar Kavramları. (n.d.). Retrieved from <http://bilgisayarkavramlari.sadievrenseker.com>
- [2] <https://ysar.net/algoritma/lz77.html>
- [3] Where Developers Learn, Share, & Build Careers. (n.d.). Retrieved from <https://stackoverflow.com>

[4] https://en.wikipedia.org/wiki/LZ77_and_LZ78

[5] <http://bilgisayarkavramlari.sadievrenseker.com/2008/10/22/c-ile-dosya-islemleri/>

[6] <https://ysar.net/algoritma/huffman-kodlamasi.html>

[7]

V. AKIŞ DİYAGRAMI

