

**VERİ YAPILARI VE ALGORİTMALAR LAB2 RAPORU**

1.

**Dersin Adı** : Veri Yapıları ve Algoritmalar

**Ödev Konusu** : Huffman Ağacı İşlemleri

A) Frekansları verilen harflerin huffman kodları bulunacaktır.

**Numara** : 12011015

**Ad Soyad** : Fatih ÇOMAK

**Tarih** : 01.11.2017

**Yürütücü** : Prof. Mehmet Yahya KARSLIGİL

2. Ödevin Amacı

Konu: Bir metin içerisinde geçen harflerin kullanım sıklıkları yardımıyla oluşturulan HTR(m) huffman ağacını kullanarak CD(nc) dizisiyle verilen kodun çözümü

A) Frekansları verilen harflerin huffman kodları bulunacaktır.

B) HTR ağacı kullanılarak verilen CD kod dizisi çözülecektir.

3. Ödevin İçeriği

* Program Kodu

#include <stdio.h>

int main()

{

int i,n,A[32],B[64],Harf[64],T,j,k,t,HTR[100],htrHarf[100];

printf("N degerini girin: ");

scanf("%d",&n);

printf("A[n] dizisini girin(kucukten buyuge): \n");

for(i=1;i<=64;i++)

{

Harf[i]=0;

}

for(i=1;i<=n;i++)

{

printf("\n%d. harfi giriniz: ",i);

scanf("\n%c %d",&Harf[i],&A[i]);

}

for(i=1;i<=(2\*n-1);i++)//2n-1 B dizisinin eleman sayisi

{

if(i<=n)

{

B[i]=A[i];

}else

{

B[i]=0;

}

}

//Toplam dizisi olusturma basladi.

i=1;

while(i<n)

{

T=B[i]+B[i+1];

i=i+2; //i indisi toplanacak elemanin yerini tutar.

k=i; //k indisi toplamin yerlesecegi yeri tutar.

while(B[k]!=0 && T>=B[k])

{

k++;

}//toplamin yerlesecegi yer bulundu.

n++;

if(B[k]!=0)

{

for(j=n;j>=k+1;j--)

{

B[j]=B[j-1];

Harf[j]=Harf[j-1];

}

}

B[k]=T;

Harf[k]=-1;

}

printf("\nToplam dizisi olusturuldu.\n");

printf("Huffman agaci olusturma basladi...\n");

for(i=1;i<=100;i++)

{

HTR[i]=0; //huffman dizisi 0'landi.

htrHarf[i]=0;

}

HTR[1]=B[n];//ilk eleman atandi.

htrHarf[1]=Harf[n];

while(n>0)

{

i=1;//huffman agacinda eleman arayacagimiz icin her defasinda 1'den baslatilir.

while(HTR[i]!=B[n])//burada da ilgili kontrol yapilir.(B[n]: B dizisinde sondan geriye dogru gelirken n'in ilgili degeri)

{

i++;

}

k=n; //k indisi de B dizisindeki n indisindeki elemanin, yanyana olan hangi iki elemanin toplami oldugunu kontrol ettigimiz indis

while((k-2)>0 && B[k-1]+B[k-2]!=B[n])//ilgili kontrol yapilir, ta ki k-2 indisi B dizisinin basina gelene kadar.

{

k--;

}

if((k-2)>0)//eger B dizisinin basina ulasilip gecilmediyse huffman agacina ilgili indisin soluna ve sagina atama yapilir.

{

HTR[2\*i]=B[k-2];

htrHarf[2\*i]=Harf[k-2];

HTR[2\*i+1]=B[k-1];

htrHarf[2\*i+1]=Harf[k-1];

}

n--;//B[n] dizisinde geriye dogru gidileceginden n degeri azaltilir.

}

printf("Huffman agaci olusturuldu.\n");

printf("\n\nHTR[] agaci:\n");

for(i=1;i<=100;i++)

{

if(htrHarf[i]>0)

{

printf("%d. dugumdeki Harf: %c , frekansi: %d\n",i,htrHarf[i],HTR[i]);

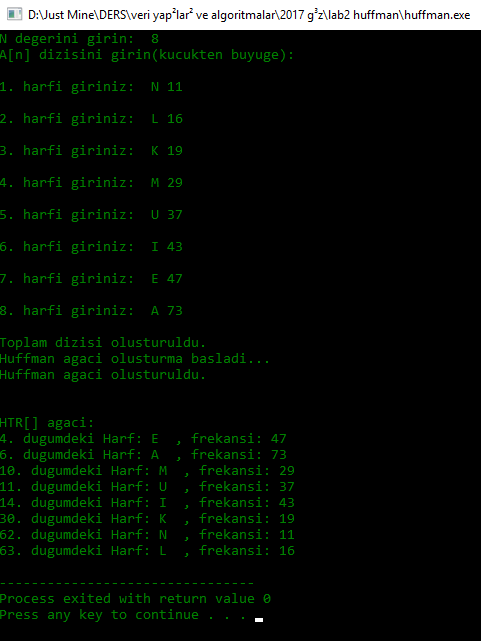
}

}

return 0;

}

* Ekran Çıktısı



* Kar Oranı Hesaplama

Giriş Bilgisi == >>

Harf ve frekans: A 73, E 47, I 43, U 37, M 29, K 19, L 16, N 11

Toplam frekans = 275

1 harf = 1 byte yani 8 bit; 275x8 = 2200 bit (sıkıştırma öncesi toplam boyut)

Huffman ağacının son halinde;

A: 1 0 2 bit 73 x 2 = 146

E: 0 0 2 bit 47 x 2 = 94

I: 1 1 0 3 bit 43 x 3 = 129

U: 0 1 1 3 bit 37 x 3 = 111

M: 0 1 0 3 bit 29 x 3 = 87

K: 1 1 1 0 4 bit 19 x 4 = 76

L: 1 1 1 1 1 5 bit 16 x 5 = 80

N: 1 1 1 1 0 5 bit 11 x 5 = 55

+\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Toplamda: 678 bit

Bu yüzden kar oranı;

(2200-678) / 2200 = **0,69** ‘dur.

4. Referanslar

\* Veri yapıları ve algoritmalar dersi notları (Prof. M.Yahya KARSLIGİL)