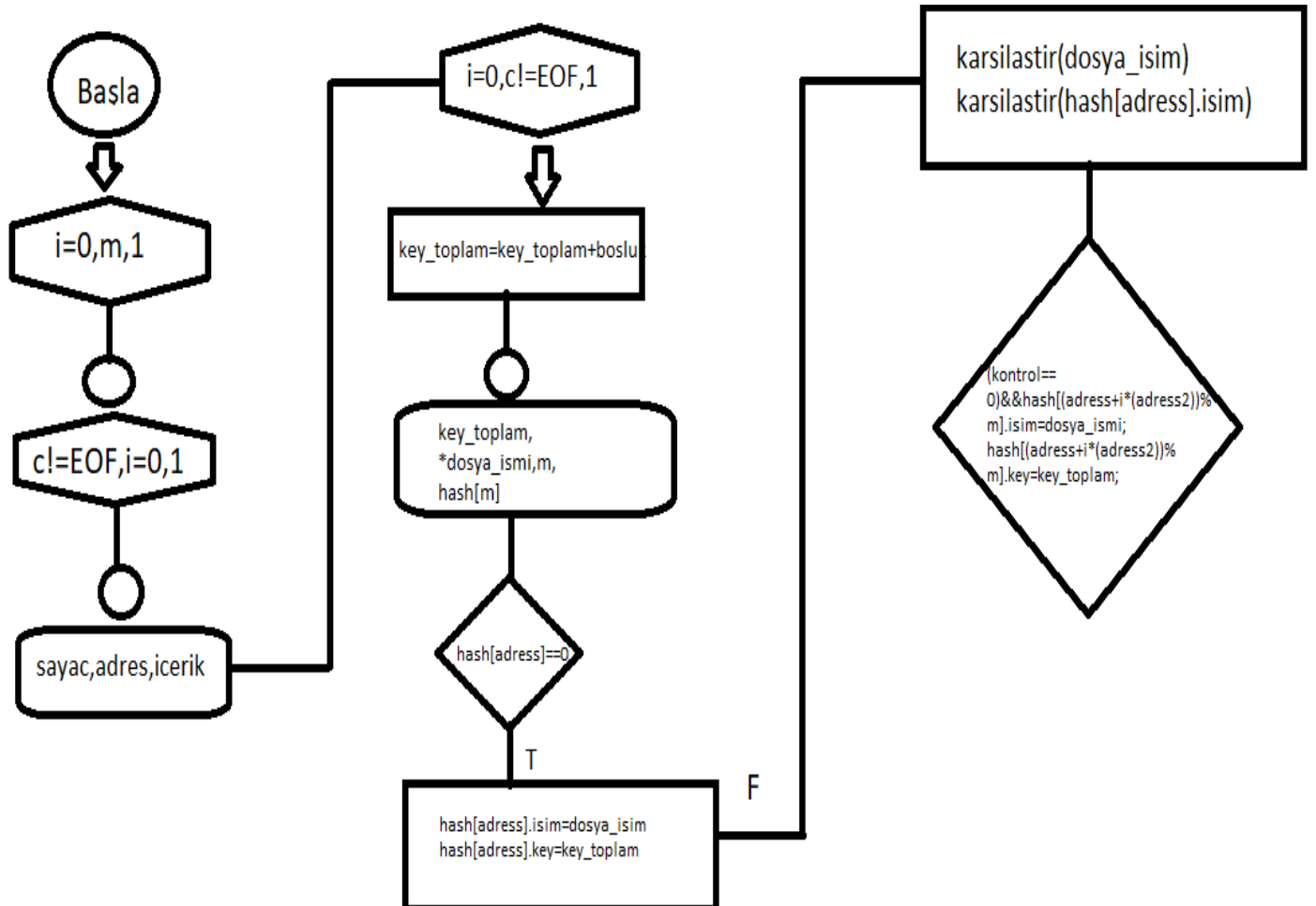


ALGORİTMA ANALİZİ 2.ÖDEV RAPORU

Yöntem Bölümü

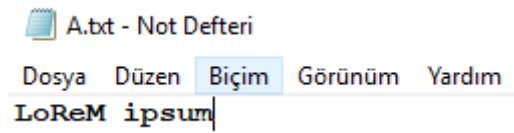
Problem zaman kaybetmeden hashing ile dosyaların birbirlerinin aynı olup olmadığını bulunması olarak tanımlanabilir. Bunun için hashte elemanlar varken ,eklenen yeni elemanlarla her birinin içeriğini kontrol etmek zor olacağından her birine belirli bir key etiketi verdiğimiz zaman ve hash e bu şekilde yerleştirdiğimiz zaman daha kolay olacaktır.

Akış Diyagramı



Uygulama Bölümü

Ödevi hazırlarken, harflerin ayırt edici özelliklerinin ASCII değerleri olduğunu düşünerek, hesaplamaları buna göre yaptım. Örnek olarak;



Bu iki kelimeyi içeren text dosyası için ASCII değer hesaplamasını şu şekilde yaptım;

L ->76 , o ->111 , R ->82 , e ->101, M ->77 , i ->105 , p-> 112 , s ->115 , u ->117 , m ->109

key=str[0] * Rn-1 + str[1] * Rn-2 + ... + str[n-1] formülüne göre , R' yi 1 almamız istendiğine göre 1 in de bütün kuvvetleri 1 olduğuna göre aslında bu harflerin değerlerinin toplamı bana key'i verir.

Burdan 76+111+82+101+77+105+112+115+117+109 = 1005 olur.Ayrıca 1 adet de boşluk (space) değeri var. Boşluğun değeri ASCII tablosuna göre 32'dir. Sonuç olarak key 1005+32=1037 'dir.

```
Okunması gereken dosya ismi : A.txt
Boşluk sayısı : 1Toplam key : 1005Key toplamı : 1037
```

Hash ' e en başta 3 txt ekledim. Buna göre tablo uzunluğu şu şekilde belirlendi ;

TabloUzunluğu=EnküçükAsalSayı >=TablodakiElemanSayısı/LoadFactor

Tablodaki Eleman Sayısı : 3 , Load Factor 0.6 alındığında; 3/0.6=5 olarak bulunur 5 bir asal sayı olduğundan eleman sayısı olarak belirlenir.

Yukarıda LoReM ipsum yazısının key'i 1037 dir. Buna göre hash tablosundaki konumu şu şekilde bulunur ;

$$1037 \% 5 = 2$$

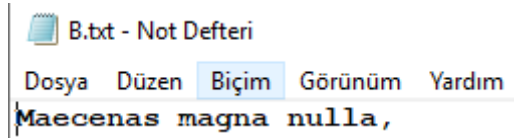
Eğer 2. Adres boşsa bu key'in 2' ye yerleşmesi gerekiyor ;

İlk önce A.txt yi okuduğumuz için adres

boş olacağından 2.ye yerleşir.

```
Okunması gereken dosya ismi : A.txt
Boşluk sayısı : 1Toplam key : 1005Key toplamı : 1037Eklenecek adres : 2
-----
Key değeri : 0      Dosya ismi : (null)
Key değeri : 0      Dosya ismi : (null)
Key değeri : 1037   Dosya ismi : A.txt
Key değeri : 0      Dosya ismi : (null)
Key değeri : 0      Dosya ismi : (null)
```

2. eklemek istediğim B.txt dosyasının içeriği ise şu şekilde ;



Key 'ini hesaplamak istersem ;

M - 77, a - 97, e -101 , c-99 , e-101 , n-110, a-97 , s-115 ,
boşluk - 32 , m-109 , a - 97 , g -103 , n- 110 , a - 97 ,
boşluk-32 , n- 110 , u-117, l- 108, l-108 , a-97, virgül - 44

Yukardaki formülü uygulayıp toplarsam ;

77+97+101+99+101+110+97+115+32+109+97+103+110+97+32+110+117+108+108+97+44 = 1961
olarak toplam key bulunur.

1961 % 5 = 1 olarak adres belirlenir .Daha önceden bu adreste değer olmadığından bu adrese yerleşir.

```
Okunmasi gereken dosya ismi : B.txt
Bosluk sayisi : 2Toplam key : 1897Key toplami : 1961Eklenen adres : 1
-----
Key degeri : 0      Dosya ismi : (null)
Key degeri : 1961   Dosya ismi : B.txt
Key degeri : 1037   Dosya ismi : A.txt
Key degeri : 0      Dosya ismi : (null)
Key degeri : 0      Dosya ismi : (null)
```

3. eklemek istediğim C.txt dosyasının içeriği ise şu şekilde ;



C.txt - Not Defteri

Dosya Düzen Biçim Görünüm Yardım

Fusce vitae lectus

Bu dosyanın hash adresini hesaplamak istersem ;

F – 70, u-117, s-115, c-99, e-101,boşluk – 32, v – 118 ,
i-105,t-116,a-97, e -101, boşluk – 32, l-108, e-101,
c-99, t-116, u-117, s-115

70+117+115+99+101+32+118+105+116+97+101+32+108+101+99+116+117+115=1759

Buna göre 1759 % 5=4 ,

4.adrese daha önce yerleştirme yapılmadı yani boş.O yüzden 4.adrese yerleşir.

```
Okunmasi gereken dosya ismi : C.txt
Bosluk sayisi : 2Toplam key : 1695Key toplami : 1759Eklenen adres : 4
-----
Key degeri : 0      Dosya ismi : (null)
Key degeri : 1961   Dosya ismi : B.txt
Key degeri : 1037   Dosya ismi : A.txt
Key degeri : 0      Dosya ismi : (null)
Key degeri : 1759   Dosya ismi : C.txt
```

Hocanın yayımladığı örnek için hash değerlerini ve yukarıda düşündüğüm yöntemle göre A.txt'nin adresi ve key'ini şöyle buldum ;

```
Okunmasi gereken dosya ismi : A.txt
Bosluk sayisi : 9Toplam key : 5824Key toplami : 6112Eklenen adres : 2
-----
Key degeri : 0      Dosya ismi : (null)
Key degeri : 0      Dosya ismi : (null)
Key degeri : 6112    Dosya ismi : A.txt
Key degeri : 0      Dosya ismi : (null)
Key degeri : 0      Dosya ismi : (null)
```

B.txt için sonuçlar şu şekilde ;

```
Okunmasi gereken dosya ismi : B.txt
Bosluk sayisi : 43Toplam key : 25417Key toplami : 26793Eklenen adres : 3
-----
Key degeri : 0      Dosya ismi : (null)
Key degeri : 0      Dosya ismi : (null)
Key degeri : 6112    Dosya ismi : A.txt
Key degeri : 26793    Dosya ismi : B.txt
Key degeri : 0      Dosya ismi : (null)
```

C.txt için sonuçlar şu şekilde ;

```
Okunmasi gereken dosya ismi : C.txt
Bosluk sayisi : 79Toplam key : 48068Key toplami : 50596Eklenen adres : 1
-----
Key degeri : 0      Dosya ismi : (null)
Key degeri : 50596    Dosya ismi : C.txt
Key degeri : 6112    Dosya ismi : A.txt
Key degeri : 26793    Dosya ismi : B.txt
Key degeri : 0      Dosya ismi : (null)
```

Yeni eklemek istediğim veri tabanımda olmayan D.txt için key değerlerinin yüzdesini alınca yani openadresing ile çakışma oluyor kontrol ediyor dosyalar aynı mı diye ve farklı olduğunu görünce double hashing ile başka bir adres buluyor ve de sonuçlar şu şekilde ;

```
Okunmasi gereken dosya ismi : D.txt
Bosluk sayisi : 110Toplam key : 70443Key toplami : 73963
-----
ADRESTEK! DOSYA ISMI : B.txt
GELEN DOSYA ISMI : D.txt
-----
Bu dosyalar farklıAdress 2 : 4
Çakisma oldu,Hash adresi : 0
-----
Key degeri : 73963      Dosya ismi : D.txt
Key degeri : 50596      Dosya ismi : C.txt
Key degeri : 6112       Dosya ismi : A.txt
Key degeri : 26793      Dosya ismi : B.txt
Key degeri : 0          Dosya ismi : (null)
-----
```

Ve son olarak E.txt dosyası C.txt dosyasıyla tamamen aynı eklememesi lazım,dosyalar aynı demesi lazım sonuçlar şu şekilde ;

```
Okunmasi gereken dosya ismi : E.txt
Bosluk sayisi : 79Toplam key : 48068Key toplami : 50596
-----
ADRESTEK! DOSYA ISMI : C.txt
GELEN DOSYA ISMI : E.txt
-----
Bu Dosyalar aynı-----
Key degeri : 73963      Dosya ismi : D.txt
Key degeri : 50596      Dosya ismi : C.txt
Key degeri : 6112       Dosya ismi : A.txt
Key degeri : 26793      Dosya ismi : B.txt
Key degeri : 0          Dosya ismi : (null)
-----
```

FONKSİYONLARIN AÇIKLANMASI

karsilastir2 : Eklenecek dosyanın karşılaştırılma işlemi için okunması

karsilastir1 : Eklenecek dosyanın ekleneceği adreste dosya bulunuyorsa bulunan dosyanın karşılaştırılma işlemi için okunması

hashAdd : Okunan dosyanın adresinin bulunması için gönderilen fonksiyon.Eğer bulunan adres boşsa ekler ama doluysa çakışan dosyayla karşılaştırılır ve aynıysa eklenmez farklıysa double hashing yöntemiyle boş olan adrese eklenir.

hashFunction : Dokümanın boşluk olmadan key değerini hesaplayan fonksiyon.

yeni_dos : Yeni eklenen dosyayı okuyan fonksiyon

dosyaOku : Veri tabanında var olan dosyaları okuyan fonksiyon

samplesDosAc : Veri tabanında bulunan dosyaları açan fonksiyon.

sayacBul : Veri tabanında bulunan dosya sayısını bulan fonksiyon.

asal : Bulunan tablo uzunluğu asal değil ise bir sonraki asal sayısı bulan fonksiyon.

dosyaSayisi : Hash tablosunun uzunluğunu belirleyen ve döndüren fonksiyon

Sonuç Bölümü

5 elemanlı bir dizim var bu durumda $\alpha=N/M$,

1 elemanım varsa 1 adımda boş adres bulurum (1/5),

2 elemanım varsa (2/5) boyutta karışıklık,

3 elemanım varsa (3/5) boyutta karışıklık,

4 elemanım varsa (4/5) boyutta karışıklık,

5 elemanım varsa (5/5) boyutta karışıklık,

Ortalama olarak key'ler in rastgele iyi dağıldığını varsayarak en kötü $\log N$ adımda bulunmuş olur. Normalde lineer olarak herbirine baksak n adımda bulmuş oluruz.