2019-2020 Güz Yarıyılı
Algoritma Analizi 2. Ödevi

Konu : Hashing Algoritması **(SON TESLİM 26 KASIM SALI 2019 23:59)**

Problem: Bir arama moturu, web sayfalarına ait dokümanları büyük bir veritabanında saklamaktadır. Bu ödevde, eklenmek istenen yeni bir dokümanın veritabanında olup olmadığını kontrol eden, eğe yoksa veritabanına ekleyen bir sistem tasarlanacaktır. Dokümanın veritabanında olup olmadığının mevcut bütün dokümanların içeriklerine tek tek bakılarak yapılması zaman alıcı bir işlemdir. Bu nedenle bu işlem **hashing** ile yapılacaktır.

Yapılacak işlemler aşağıdaki gibidir:

1. İşlem yapılacak dosyaları bir *directory* altında toplayınız. Bu dosyaların isimlerini ***samples.doc*** isimli bir dosyaya yazınız.
2. **Hash tablosunu oluşturma :** ***samples.doc*** dosyasını okumak için açarak, sırası ile bu dosyada adı olan her dosyayı okumak için açınız. Her dosyadaki karakterleri hash fonksiyonundan geçirerek dosyanın **hash sayısını** hesaplayınız. Hash tablosunda hash sayısının olduğu adres :
   a. **boş ise** bu adrese dokümanın adını yazınız.
   b. bu adres **dolu ise** aşağıda açıklandığı şekilde **double hashing** yöntemi ile dokümanın adını hash tablosunun uygun adresine yerleştiriniz.
3. **Yeni bir dokümanın veritabanına eklenmesi :** Yeni bir dokümanın veritabanında olup olmadığına karar vermek ve eğer yoksa eklemek için aşağıdaki işlemler yapılacaktır:
   a. Dokümanı okuyunuz.
   b. Dokümanı hash fonksiyonundan geçirerek hash sayısını elde ediniz.
   c. Hash tablosunda hash sayısının bulunduğu adres :
      i. **boş ise** bu adrese dokümanın adını yazınız. ***samples.doc*** dosyasına bu dokümanın adını da ekleyiniz. Dokümanı diğer dokümanların olduğu *directory* altına yerleştiriniz.
      ii. bu adres **dolu ise**, bu adreste bulunan dokümanın adını okuyunuz ve doyayı okumak için açınız. Bu doküman ile yeni dokümanın içeriklerinin aynı olup olmadığını karşılaştırınız. Karşılaştırma işlemi için C'nin **strcmp** fonksiyonunu kullanabilirsiniz. Eğer **doküman içerikleri aynı ise** işleminiz tamamlanmıştır. Doküman **içerikleri farklı ise** aşağıda açıklandığı şekilde **double hashing** yöntemi ile tablodaki bir sonraki adresi hesaplayarak aynı işlemleri tekrarlayınız. Eğer bu dokümanla aynı içerikte bir doküman bulunamadın boş adrese gelinirse bu adrese dokümanın adını yazınız. ***samples.doc*** dosyasına bu dokümanın adını da ekleyiniz. Dokümanı diğer dokümanların olduğu *directory* altına yerleştiriniz.

Hash Tablosunu Oluşturma:

1. Hash tablosunu oluştururken *openaddress,* çakışma problemini çözmek için *double hashing* yöntemleri kullanılacaktır. Buna göre:
   h(key,i) = [h1(key) + i*h2(key)] mod M
   h1(key) = key mod M
   h2(key) = 1 + (key mod MM)

2. Tablo uzunluğunu gösteren **M** değerinin belirlenmesi için aşağıdaki bağıntı kullanılacaktır :
   TabloUzunlugu = EnküçükAsalSayı >= TablodakiElemanSayısı / LoadFactor
   LoadFactor = 0.6 alınız. Bu işlemi elle yapınız. Tablo uzunluğunu bulmak için program yazmayınız. h2 fonksiyonunda **MM = M -1** alınız.

3. Her dosyanın büyük ve küçük harflerden oluşan yazılar olduğu kabul edilmiştir. Küçük harflerden oluşan, bir str kelimesi için i=0..n-1 arası değişirken hash fonksiyonuna verilecek **key** sayısı R bir asal sayı seçilerek aşağıdaki gibi hesaplanır:
   key = $\texttt{str[0]} * R^{n-1} + \texttt{str[1]} * R^{n-2} + \ldots + \texttt{str[n-1]}$

.

Teslim Edilecekler: Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayınız.
Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız.

1. **Yöntem** bölümünde problemi kısaca anlatıp, algoritmanıza ait **akış diagramını** çiziniz.
2. **Uygulama** bölümünde önerdiğiniz algoitmanın analizini yapınız. Analiz olarak sonucun ekran çıktısını vermeniz değil, derslerde yapıldığı gibi küçük bir örnek üzerinde ana değişkenlerin değişimini ve çözümün elde edilişini adım adım göstermeniz gerekmektedir.
    a. Küçük ve büyük harflerden oluşan uzun olmayan 10 dosyanın hash adreslerini hesaplayarak hash tablosuna yerleştiriniz. Dosya içeriklerini oluştururken 3 dosya için aynı hash adresini oluşturarak çakışma olmasını sağlayınız. Kolayca çakışma olmasını sağlamak için yukarıda stringi sayıya dönüştürerek key değeri elde ederken kullanılan bağıntıda R=1 alınız.
    b. Yeni dosya ekleme için bir adet veritabanında olan dosya için, bir adet de veritabanında olmayan dosya için algoritmanızın çalışmasının ana adımlarına ait değişimi gösteriniz.
3. **Sonuç** bölümünde tasarladığınız hashing algortitmasının **en kötü durum** karmaşıklığını hesaplayarak performansını yorumlayınız.
4. Algoritmanızın **C dilinde** programını hazırlayarak dokümana ekleyiniz.

Teslim İşlemleri:
Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak 26 Kasım 2019 saat 23:59'a kadar adresi https://forms.gle/K8wjU2d1sZBaACdJ6 üzerinden HW2_OgrenciNumarasi.rar dosyasını yükleyiniz.

Ödevler **27 Kasım 2019** günü yapılacak laboratuvarda gösterilecektir. Ödev teslim ve sunum planı için Arş. Grv. Ahmet Elbir'in Avesis sayfasını takip ediniz.

Laboratuvar Sunumu: Programınızın çalışmasını laboratuvar esnasında size verilecek olan bir örnek üzerinde göstermeniz istenecektir.

Değerlendirme: Ödeviniz aşağıdaki gibi değerlendirilecektir:

Algoritma Tasarımı ve Programın Çalışması: (%70)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Programda gerekli alt modüller belirlenerek her modül ayrı fonksiyon olarak yazılmalıdır.
4. Program hatasız çalışmalıdır.
5. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%30)

1. Raporun ilk sayfasında, dersin adı, öğrencinin ad, soyad ve numarası, ödev konusu bilgileri yer almalıdır.
2. Kaynak kodda değişken deklerasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
3. Değişken ve fonksiyon(veya metod) isimleri anlamlı olmalıdır.
4. Her fonksiyonun (veya metodun) yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
5. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmadır.
6. Gereksiz kod tekrarı olmamalıdır.
7. Kaynak kodun formatı düzgün olmalıdır.

******* Teslim Edilecekler**

a. HW#_OgrenciNumarasi.rar (Örn: HW2_15011001.rar)
i. OgrenciNumarasi.pdf (Örn: 15011001.pdf)
ii. OgrenciNumarasi.c (Örn: 15011001.c)