

2018-2019 Bahar Yarıyılı

Algoritma Analizi

Ödev – 2

Ders Yürütücüsü: Dr. Öğr. Üyesi M. Amaç GÜVENSAN

Konu : Hashing

Problem: Bu ödevde harf yazım kontrolü (spell checker) işlemi yapan bir sistem tasarlanacaktır. Sistem, dışardan verilen bir kelimenin sözlükte olup olmadığını kontrol edecek, kelimeyi sözlükte bulursa ‘kelime doğru yazılmıştır’ mesajı verecek, kelimeyi sözlükte bulamazsa sözlükte bulunan kelimelerden doğru kelime önerilerini verecektir. Bu ödevde hash tablosu oluşturulacak sözlük için ekte verilen **smalldictionary.txt** dosyası kullanılacaktır.

Ödev üç ana bölümden oluşacaktır :

1. **Hash Tablosu Oluşturma** : Doğru yazılmış kelimelerin saklandığı sözlük dosyasındaki(ekteki smalldictionary) kelimeler hash tablosuna yerleştirilir. Bu işlem bir defa yapılacaktır.
2. **Verilen Kelimeyi Hash Tablosunda Arama** : Yazım kontrolünden geçirilmek için dışardan okunan kelime hash tablosunda aranır. Eğer kelime sözlükte bulunamazsa, yanlış kelime yerine sözlükte bulunan benzer kelimeler önerilir.
3. **Yanlış Kelime Yerine Kelime Önerme** : Yanlış kelime yerine kelime önerme işlemi için her seferinde verilen kelimedeki tek harfi değiştiren basit bir yöntem kullanılacaktır. Bu yöntemde kelimedeki her harf sırası ile ‘a’ dan ‘z’ ye harflerle değiştirilerek elde edilen yeni kelimenin hash tablosunda olup olmadığına bakılacaktır. Eğer elde edilen yeni kelime hash tablosunda varsa yanlış kelime yerine önerilen kelimelerden biri olarak ekrana yazdırılacaktır.

Örneğin yanlış kelime **sate** olsun. Bunun yerine sistemin önerebileceği kelimeler **late, same, save, ...** olabilir.

Hash Tablosu aşağıdaki şartları sağlayacak şekilde oluşturulmalıdır:

1. Tabloya yerleştirilecek büyük harflerden oluşan bir kelimenin sayı karşılığını aşağıdaki gibi hesaplayınız.

```
for(i=key=0;i < length(str); i++) key += (str[i] - 'a')*26*i ;
```

2. Hash tablosunu oluştururken **OpenAddress, Çakışma(collusion)** problemini çözmek için **Linear Probing** yöntemini kullanınız. Hash fonksiyonu olarak bölme (division) yöntemini kullanınız. Hash fonksiyonlarını aşağıdaki bağıntıları kullanarak belirleyiniz:

$$h(\text{key}, i) = (h1(\text{key}) + i * \text{offset}) \% m$$

$$h1(\text{key}) = \text{key} \% m$$

$$\text{offset} = R - (\text{key} \% R)$$

3. **Tablo uzunluğunu gösteren m değerinin belirlenmesi için** aşağıdaki bağıntıyı kullanınız :

$$m = \text{EnkçükAsalSayı} \geq \text{TablodakiElemanSayısı} / \text{LoadFactor}$$

R değerini m'den küçük ilk asal sayı olarak alınız.

4. LoadFactor'u 0.70 alınız. Tablo uzunluğunu belirleme işlemini elle yapınız, bunu için program yazmayınız.

Ödevinizi **NORMAL** veya **DEBUG** mod olmak üzere 2 modda çalışacak şekilde yazın. Program çalıştırılırken parametre olarak NORMAL mod verilirse ekranda sadece yukarıda oyun ile ilgili istenilenlere ait mesajları yazdırın. Programın DEBUG modda çalıştırılması istenirse hash tablosunda kelime arama işleminin her adımını da ekranda yazdırın. DEBUG modda **m, R, her kelime için key, h1 ve offset değerleri ve kelimeyi bulmak için her adımda kaçınıcı adrese** bakıldığı yazılmalıdır.

Program çalışmaya başladığında hash tablosu oluşturulmasının istenip istenmediğini kullanıcıya sormalı, eğer oluşturması isteniyorsa hash tablosunu oluşturmalı, istenmiyorsa mevcut hash tablosunun saklı olduğu dosya ismini sormalı ve dosya var ise bu dosyadan hash tablosunu okumalıdır.

1. Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız.
2. Yöntem bölümünde problemi ve gerçekleştirdiğiniz çözümü kısaca anlatıp, algoritmanıza ait pseudo code'ünü yazınız.

3. Uygulama bölümünde aşağıdaki 40 kelime için kelimelerin hash tablosunda arama adım

sayısını hesaplayınız. Ortalama arama sayısının ikili arama yönteminden daha hızlı olup olmadığını değerlendiriniz. FACE FACT GOLE GOLF HOLE HOLK BISK BITE DAME DAMN NIMS NINE PAIN PAIR LIVE LOAD YARD YARE ZERO ZEST FACX FACZ GOLT GOLZ HOLF HOLR BIST BITF DAMF DAMZ NIMT NINF PAIY PAIZ LIVT LOAR YARF YART ZERT ZESZ

Ödev Teslimi ile ilgili önemli bilgiler:

Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak **24.04.2019 saat 06:59'a** kadar <https://forms.gle/Me2A2JMaUkALcU1FA> adresi üzerinden **HW2_OgrenciNumarasi.rar** dosyasını yükleyiniz.

- Ödev kontrolleriniz **24.04.2019 Çarşamba günü** dersin laboratuvar saatinde yapılacaktır. Detaylı bilgi için **Arş. Grv. Ahmet ELBİR'in** avesis.yildiz.edu.tr adresindeki sayfasını takip ediniz.

Teslim Edilecekler:

1. Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız. **Yöntem** bölümünde problemi ve çözüm için önerdiğiniz yöntemi kısaca anlatıp rekürsif fonksiyonunuzun durma ve rekürsif olarak çağırılma koşullarını belirtiniz.
2. **Uygulama** bölümünde son taşa ulaşılabilen ve ulaşılamayan 4 ayrı örnek üzerinde algoritmanızın çalışmasını adım adım gösteriniz.
3. Algoritmanızın **C** dilinde programını hazırlayarak dokümana da ekleyiniz.
4. **Teslim Edilecekler**
 - a. HW#_OgrenciNumarasi.rar (Örn: HW2_15011001.rar)
 - i. OgrenciNumarasi.pdf (Örn: 15011001.pdf)
 - ii. OgrenciNumarasi.c (Örn: 15011001.c)

Değerlendirme

Algoritma Tasarımı ve Programın Çalışması: (%80)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Program hatasız çalışmalıdır.
4. Programın çalışması sırasında, konuyu bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%20)

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