2014-2015 Güz Yarıyılı
Algoritma Analizi 3. Ödevi

Konu : Hashing

Problem: Anagram oyunu

Anagram Yunan dilinde, harfleri altüst etmek anlamına gelen *anagramma* kelimesinden gelir. **Bir kelimenin anagramı sözcüğün harf sayısını bozmadan harflerinin yeri değiştirildiğinde ortaya çıkan kelimedir.** Örneğin 'askı' kelimesi 'kısa' kelimesinin anagramıdır. Bu ödevde tasarlanacak oyunun çalışması aşağıdaki gibidir:

1. Oyun için http://members.ozemail.com.au/%7Erjackman/4fours.html adresindeki 4 harfli kelimelerden oluşan sözlüğü kullanın.
2. Oyuncuya bu sözlükte bulunan bir kelimeyi vererek bu kelimenin anagramlarını üretmesini isteyin.
3. Oyuncu sorulan her kelime için istediği kadar çok kelime üretebilir. Oyuncunun doğru ürettiği her kelime için 5, yanlış ürettiği her kelime için -5 puan verilerek her turun sonunda aldığı puan hesaplanır.
4. Eğer verilen kelime için ürettiği kelimeler bitti ise bu tur için toplam puanı ve oyunda o ana kadar aldığı toplam puanlar hesaplanarak ekrana yazdırılır. Oyuncu devam etmek isterse yeni bir kelime sorulur. Devam etmek istemezse oyun sonlandırılır.

Ödev 3 ana bölümden oluşmaktaktır :

1. **Hash tablosu oluşturma :** Yukarıdaki adresteki kelimeleri bir dosyaya kopyalayınız. Daha sonra bu kelimeleri bir hash tablosuna yerleştiriniz. Bu işlem bir defa yapılacaktır.
2. **Oyuncunun verdiği kelimenin anagram olup olmadığının kontrolü :**
   Oyuncunun verdiği kelimenin anagram olup olmadığı kontrol edilir.
3. **Oyuncunun verdiği kelimeyi hash tablosunda arama :** Oyuncunun verdiği kelime **anagram kuralını sağlıyorsa** bu kelimenin hash tablosunda olup olmadığı kontrol edilir.

Hash Tablosu aşağıdaki şartları sağlayacak şekilde oluşturulacaktır:

1. Tabloya yerleştirilecek büyük harflerden oluşan bir kelimenin sayı karşılığı *Horner Kuralı* kullanılarak aşağıdaki gibi hesaplanır.

```
for(i=key=0; i < length(str);  i++)
        key = 61 * key + str[i];
```

2. Hash tablosunu oluştururken *openaddress*, çakışma(*collusion*) problemini çözmek için *double hashing* yöntemini kullanınız. Hash fonksiyonu olarak bölme (*division*) yöntemini kullanınız. Hash fonksiyonlarını aşağıdaki bağıntıları kullanarak belirleyiniz:
   h(key, i) = (h1(key) + i*h2(key)) % m
   h1(key) = key % m
   h2(key) = 1 + (key % m2)
   h2 fonksiyonundaki m2 değerini m2 = m-1 olarak alınız.
3. Tablo uzunluğunu gösteren *m* değerinin belirlenmesi için aşağıdaki bağıntıyı kullanınız :
   TabloUzunlugu = EnküçükAsalSayı >= TablodakiElemanSayısı / LoadFactor

   LoadFactor'u 0.60 alınız. Tablo uzunluğunu belirleme işlemini elle yapınız, bunu için program yazmayınız.

Ödevin çalıştırılması :

1. Ödevinizi **NORMAL** veya **DEBUG** mod olmak üzere 2 modda çalışacak şekilde yazın.

   a.) NORMAL Mod :
   Program çalıştırılırken parametre olarak NORMAL mod verilirse ekranda sadece yukarıda oyun ile ilgili istenilenlere ait mesajları yazdırın.

   b.) DEBUG Mod:
   Programın DEBUG modda çalıştırılması istenirse hash tablosunda kelime arama işleminin her adımını da ekranda yazdırın. Örneğin kullanıcının önerdiği kelime HOST olsun. h1("HOST") = 70 h2("HOST") = 123 olduğunu ve hash tablosu oluşturulurken "HOST" kelimesinin 3. denemede boş bulunan yer olan 316. adrese (70 + 123 + 123) yerleştiğini varsayalım. Bu durumda **NORMAL moddaki mesajlara ek olarak** DEBUG modda aşağıdaki mesajlar da verilmelidir:

```
h1("HOST") = 70
h2("HOST") = 123
HOST kelimesi 70. adreste bulunamadı.
HOST kelimesi 193. adreste bulunamadı.
```

HOST kelimesi 316. adreste bulundu.

2. Kelime sayısı çok olduğu için laboratuvarda ödevinizin çalışmasını gösterirken hash tablosunu oluşturursanız uzun zaman alabilir. Bu nedenle 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.

Teslim Edilecekler: Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayınız.

1. Yaptığınız çalışmayı yöntem ve uygulama bölümlerinden oluşan bir raporda anlatınız.
2. **Yöntem** bölümünde problemi ve gerçekleştirdiğiniz çözümü kısaca anlatıp, algoritmanıza ait pseudo code'unu yazınız.
3. **Uygulama** bölümünde aşağıdaki 40 kelime için kelimelerin hash tablosunda aranma adım sayısını hesaplayınız. Ortalama aranma 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
```

4. Algoritmanızın C dilinde programını hazırlayarak dokümana ekleyiniz.

Teslim Tarihi:
Ödevler **22 Aralık 2014** haftası yapılacak labaratuvarda gösterilecektir. Rapor teslimi için yapılması gereken işlemler laboratuvar sorumlusu asistanının sayfasında açıklanmıştır.

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

Değerlendirme:Ödeviniz aşağıdaki gibi değerlendirilecektir:

Algoritma Tasarımı ve Programın Çalışması: (%80)

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: (%20)

1. Raporun kapak 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.