

2020-2021 Güz Yarıyılı

Algoritma Analizi

Ödev – 3

Ders Yürütücüleri

Doç. Dr. M. Elif KARSLIĞİL

Dr. Öğr. Üyesi M. Amaç GÜVENSAN

Konu : Dinamik Programlama

Problem 1: A ve B takımları, içlerinden biri n galibiyet alana kadar maç yapacaklardır. A takımının bir maçı kazanma olasılığı her maç için p, kaybetme olasılığı ise 1-p'dir. Dolayısıyla beraberlik ihtimali mevcut değildir. A takımının seriyi kazanmak için i tane daha maç kazanması, B takımının da seriyi kazanmak için j tane maç kazanması gereken durumda A'nın seriyi kazanma olasılığı $P(i,j)$ 'dir.
(30 Puan)

- $P(i,j)$ 'yi kullanarak **dinamik programlama** ile algoritma tasarlayabilmek için **rekürans bağıntısını** yazınız.
- A takımının bir maçı kazanma olasılığının 0.6 olduğu durumda 7 maçlık seride (4 alan kazanır) **A takımının kazanma ihtimalini** hesaplayınız. (Dinamik programlama yaklaşımını kullanınız.)
- Belirtilen problemi çözen dinamik programlama yaklaşımını kullanan algoritmanın **kodunu C dilinde** yazınız.

İpucu: A takımının bir maçı kazanma olasılığının 0.4 olduğu durumda 7 maçlık seride (4 alan kazanır) A takımının kazanma ihtimali 0.29'dur.

Problem 2: Sorgulanan bir cümlede yanlış yazılmış kelimeler varsa bu kelimelerin yerine doğru kelimeler öneren bir sistem tasarlanacaktır. (70 Puan)

Örnek :

Kullanıcı: It is coold

Bilgisayar: “coold” is not in the dictionary. Did you mean: “cool “ or “cold”

Kullanıcı: cold

Bilgisayar: It is cold

İşlem Adımları: Problemde izlenmesi gereken işlemler aşağıdaki gibidir:

- Ödevde kullanılacak sözlüğü hazırlamak için **smalldictionary.txt** dosyasındaki sözlüğe ait kelimeler *sözlük tablosuna (hash tablosu)* yerleştirilir. Bu işlem bir defa yapılacaktır.
- Verilen cümledeki her kelime *sözlük tablosunda* aranır. Eğer kelime *sözlük tablosunda*
 - varsa** kelime doğrudur. O kelime için işlem tamamlanır.
 - yoksa**, *hatalı kelime* tablosunda aranır.
- Kelime *hatalı kelime tablosunda (hash tablosu)*
 - yoksa** kelimenin sözlük tablosundaki her kelimeye mesafesi *Levenshtein Edit Distance* yöntemi ile hesaplanarak, önce mesafesi **1** olan sözlük kelimeleri, eğer mesafesi 1 olan yoksa mesafesi **2** olan sözlük kelimeleri kullanıcıya önerilir. Daha büyük mesafeli kelimeler önerilmez. Hatalı kelime ve kullanıcının önerilen kelimeler içinden seçtiği doğru

kelime, *hatalı kelime tablosuna* eklenir.

- b. **vars** hatalı kelime tablosunda bu hatalı kelime bulunur ve bu kelime için daha önce önerilip kullanıcı tarafından kabul edilmiş olan doğru kelime kullanıcıya önerilir.

4. *Sözlük* ve *hatalı kelime* hash tabloları *openaddress*, çakışma problemini çözmek için *double hashing* yöntemleri ile oluşturulacaktır. Tabloları oluştururken ve arama yaparken 2. ödevde yazdığınız fonksiyonlardan faydalanabilirsiniz.

Bonus : Verilen $A[n]$ ve $B[m]$ stringleri arasındaki *edit distance* hesabını sonuna kadar yapmayıp, A ve B stringleri arasındaki mesafe k değerinden (bu ödev için mesafesi 2'den büyük olan sözlük kelimeleri kullanılmayacağı için $k=2$ 'dir) büyük olacağı **kesin olduğu** adımda işlemi bitirenler **ek 20 puan** alacaktır. Verilen $A[n]$ ve $B[m]$ stringleri ve k değeri için değişim matrisinde ana diagonalde $[0,0]$ 'dan en fazla kaç adım ilerlenmesi yeterlidir? Kısaca açıklayınız. **(20 Puan)**

Ödev Teslimi ile ilgili önemli bilgiler:

Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak **20.12.2020 saat 23:59'a** kadar online.yildiz.edu.tr adresi üzerinden **OgrenciNumarasi.rar** dosyasını yükleyiniz.

Teslim Edilecekler:

1. Algoritmanızın **C** dilinde programını hazırlayarak dokümana da ekleyiniz.
2. Her algoritma için farklı durumlarını gösteren ekran çıktılarını dokümanda paylaşınız.
3. Teslim Edilecekler
 - a. HW#_OgrenciNumarasi.rar (Örn: 15011001.rar)
 - i. OgrenciNumarasi_QuestionNo.pdf (Örn: 15011001_1.pdf)
 - ii. OgrenciNumarasi_QuestionNo_Part.c (Örn: 15011001_1.c)
 - iv. OgrenciNumarasi_QuestionNo.pdf (Örn: 15011001_2.pdf)
 - v. OgrenciNumarasi_QuestionNo.pdf (Örn: 15011001_2.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.
8. **ÖRNEK ÇIKTI EKLANLARINI mutlaka rapora ekleyiniz.**