

## 2018-2019 Güz Yarıyılı Algoritma Analizi 2. Ödevi

### Konu : Dinamik Programlama

**Problem:** Bu ödevde verilen bir tanıtım ile ilgili kelime grubunu bir reklam afişine her satırda en az boşluk kalacak şekilde yerleştirmenizi sağlayacak bir algoritma tasarlamanız beklenmektedir. Reklam afişindeki her satıra en fazla M karakter yazılabilmektedir. Tanıtım sloganları uzunluğu  $l_1, l_2, \dots, l_n$  olan farklı uzunlukta kelimelerden oluşmaktadır.

Her kelime arasında 1 boşluk bırakılarak  $i < j$  olmak üzere bir satıra i. kelimedenden j. kelimeye kadar kelime yazılırsa, satır sonunda kalan boşluk sayısı aşağıdaki gibi hesaplanır.

$$\text{space}[i,j] = M - j + i \sum_{k=i..j} l_k$$

Bir satıra i. kelimedenden j. kelimeye kadar kelime yazıldığında bu satırdaki boşlukların bedeli aşağıdaki gibi hesaplanmaktadır.

$$\text{spaceCost}[i,j] = \text{space}[i,j]^3$$

Burada eğer i. kelimedenden j. kelimeye kadar olan kelimeler **satıra sığmıyorsa** yani  $\text{space}[i,j] < 0$  ise bu işlemin **yapılmaması** gerektiği için  $\text{spaceCost}[i,j]$ 'nin değeri  $\infty$ , satır sonunda hiç **boşluk kalmıyorsa 0'dır**. Satır sonunda **boşluk varsa**, eklenen her boşluk karakterinin etkisini arttırmak için **bedel boşluk sayısının küpü** ( $\text{space}[i,j]^3$ ) olarak kabul edilmiştir. **Son satırın boşluk bedeli** hesaplanmayacağı için son satırın boşluk değeri **0** olarak kabul edilir.

1. kelimedenden j. kelimeye kadar olan kelimeleri satırlara yerleştirmenin bedeli  $\text{cost}[j]$  ile gösterilirse, o anda bulunulan satırda i..j arası kelimeler varken  $\text{cost}[j]$  aşağıdaki gibi hesaplanır.

$$\text{cost}[j] = \text{cost}[i-1] + \text{spaceCost}[i,j]$$

Bir satırın **hangi kelime ile başlaması durumunda bedelin en düşük olacağını belirlemek için** 1. kelimedenden j. kelimeye kadar olan kelimelerin **minimum bedeli**  $\text{cost}[j]$  değeri aşağıdaki gibi hesaplanır.

$$\begin{aligned} \text{cost}[j] &= 0 & \text{if } j=0 \\ \text{cost}[j] &= \min_{1 \leq i \leq j} (\text{cost}[i-1] + \text{spaceCost}[i,j]) & \text{if } j>0 \end{aligned}$$

**İşlem Adımları:** Verilen N uzunluklu bir yazıyı her satıra en fazla M karakter yazılacak şekilde satırlara yerleştirmek için aşağıdaki işlemleri yaparak algoritmanızı gerçekleştiriniz.

1. Bir satıra i. kelimedenden j. kelimeye kadar olan kelimelerin yazılması halinde oluşacak boşluk sayısını gösteren **space[i,j]**'yi  $1 \leq i \leq j \leq n$  değerleri için hesaplayın.
2. Bir satıra i. kelimedenden j. kelimeye kadar olan kelimeler yazıldığında bu satırdaki boşlukların bedeli olan **spaceCost[i,j]**'yi  $1 \leq i \leq j \leq n$  değerleri için hesaplayın.
3. 1. kelimedenden j. kelimeye kadar olan en düşük bedeli gösteren **cost[j]**'yi hesaplayıp, bulunan minimum  $\text{cost}[j]$  değerine bağlı olarak her satırın kaçınıcı kelime ile başladığını **lineAdr** dizisinde dizide saklayınız.
4. Sonuçta **yazıyı satır satır ekrana yazdırınız**.
5. **Toplam boşluk bedelini yazdırınız**.

### **Ödev raporu**

1. Yaptığınız çalışmayı **yöntem, uygulama ve sonuç** bölümlerinden oluşan bir doküman hazırlayarak anlatınız.  
**Yöntem** bölümünde problemi kısaca anlatınız.  
**Uygulama** bölümünde Arş. Grv. Ahmet Elbir'in sayfasında vereceği örnek bir yazı için
  - a.) space ve spaceCost matrislerinin, cost ve lineAdr dizilerinin alacağı değerleri hesaplayıp gösteriniz.
  - b.) Yazıyı satırlar halinde yazdırınız ve toplam boşluk sayısını veriniz.
2. Algoritmalarınızın **C dilinde** programını hazırlayarak dokümana ekleyiniz.

### **Teslim İşlemleri:**

Ödevler **6 Aralık 2018** günü yapılacak laboratuvarda gösterilecektir. Ödev teslimi için Arş. Grv. Ahmet Elbir'in 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. Rapor yukarıda istenilen bilgileri içermelidir.
3. Kaynak kodda değişken deklarasyonu yapılırken her değişken tek satırda tanımlanmalı, tanımın yanına değişkenin ne için kullanılacağı açıklama olarak yazılmalıdır.
4. Değişken ve fonksiyon(veya metod) isimleri anlamlı olmalıdır.
5. Her fonksiyonun (veya metodun) yaptığı iş, parametreleri ve dönüş değeri açıklanmalıdır.
6. Gerekli yerlerde açıklama satırları ile kodda yapılan işlemler açıklanmalıdır.
7. Gereksiz kod tekrarı olmamalıdır.
8. Kaynak kodun formatı düzgün olmalıdır.