

**Manisa Celal Bayar Üniversitesi Yazılım Mühendisliği Bölümü**  
**YZM 3207- Algoritma Analizi ve Tasarım**

**Örnek Arasınav Soruları**

Güz 2017

<b>Adı ve Soyadı</b>	<b>YANIT ANAHTARI</b>	<b>Öğrenci Numarası</b>	
<b>Grubu</b>		<b>İmza</b>	
<b>Tarih</b>		<b>Not</b>	/100

**Soru#1 (20 puan)**

Aşağıda verilen algoritmaya dayalı olarak seçeneklerdeki soruları yanıtlayınız:

```
ALGORİTMA Enigma(A[0..n-1,0..n-1])
//Girdi: A[0..n-1,0..n-1] reel sayılardan oluşan bir matris
for i ← 0 to n-2 do
    for j ← i+1 to n-1 do
        if A[i,j] ≠ A[j,i]
            return false
return true
```

- a) Algoritmanın neyi hesapladığını belirleyiniz.
- b) Algoritmanın temel operasyonu nedir?
- c) Algoritmanın temel operasyonu kaç kez işletilmektedir?
- d) Algoritmanın etkinlik sınıfını belirleyiniz.
- e) Algoritma ile aynı işlemi gerçekleştiren daha etkin bir algoritma tasarlanabilir mi? Eğer mümkünse algoritmanızı sözde kod biçiminde belirtiniz.

- a) Algoritma girdi matrisi simetrik ise "True" aksi takdirde "False" değeri döndürmektedir.
- b) İki matris elemanının karşılaştırılması.

$$\sum_{i=l}^u 1 = \underbrace{1+1+\dots+1}_{u-l+1 \text{ times}} = u - l + 1 \quad (l, u \text{ are integer limits, } l \leq u); \quad \sum_{i=1}^n 1 = n$$

$$C_{\text{worst}}(n) = \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} 1 = \sum_{i=0}^{n-2} [(n-1) - (i+1) + 1]$$

$$= \sum_{i=0}^{n-2} (n-1-i) = (n-1) + (n-2) + \dots + 1 = \frac{(n-1)n}{2}.$$

c)

$$C_{\text{worst}}(n) \in \Theta(n^2) \quad (\text{or } C(n) \in O(n^2)).$$

d)

- e) Algoritma optimaldir. Belirtilen problemi çözen herhangi bir algoritmanın en kötü durumda matrisin üst bölümündeki  $(n-1)n/2$  elemanı, matrisin alt bölümündeki  $(n-1)n/2$  eleman ile karşılaştırmak zorundadır.

**Soru#2 (20 puan)**

a)  $M(n) = 3M(n-1) + 2, n > 1$  ve  $M(1) = 2$  için.

Yukarıda verilen özyineleme bağıntısını **geriye doğru yerine koyma** (*backward substitution*) yöntemi kullanarak çözünüz.

$$\begin{aligned}
 M(n) &= 3M(n-1) + 2 \\
 &= 3[3M(n-2) + 2] + 2 = 3^2M(n-2) + 3 \cdot 2 + 2 \\
 &= 3^2[3M(n-3) + 2] + 3 \cdot 2 + 2 = 3^3M(n-3) + 3^2 \cdot 2 + 3 \cdot 2 + 2 \\
 &= \dots \\
 &= 3^i M(n-i) + 2(3^{i-1} + 3^{i-2} + \dots + 1) = 3^i M(n-i) + 3^i - 1 \\
 &= \dots \\
 &= 3^{n-1} M(1) + 3^{n-1} - 1 = 3^{n-1} \cdot 2 + 3^{n-1} - 1 = 3^n - 1.
 \end{aligned}$$

b)  $C_w(n) = C_w(n-1) + n - 1, n > 1$  ve  $C(1) = 0$  için.

Yukarıda verilen özyineleme bağıntısını **geriye doğru yerine koyma** (*backward substitution*) yöntemi kullanarak çözünüz.

$$\begin{aligned}
 C_w(n) &= C_w(n-1) + n - 1 \\
 &= [C_w(n-2) + n - 2] + n - 1 \\
 &= [C_w(n-3) + n - 3] + n - 2 + n - 1 \\
 &= \dots \\
 &= C_w(n-i) + (n-i) + (n-i+1) + \dots + (n-1) \\
 &= \dots \\
 &= C_w(1) + 1 + 2 + \dots + (n-1) = 0 + (n-1)n/2 = (n-1)n/2.
 \end{aligned}$$

**Soru#3 (20 puan)**

Birleştirme sıralaması (Mergesort), sıralanacak olan diziyi ikiye bölerek kalan parçalara inene dek sürekli olarak ikiye bölen bir sıralama algoritmasıdır. Birleştirme sıralamasında dizinin iki yerine üç parçaya ayrılması üzerinden algoritma işletildiği takdirde “üçlü birleştirme sıralama” adını almaktadır.

a) Üçlü birleştirme sıralama algoritmasına ilişkin sözde kodu yazınız. Sıralı iki B ve C dizisinin sıralı bir A dizisine birleştirilmesinin  $\text{Merge}(B, C, A)$  algoritması kullanılarak gerçekleştirildiğini varsayınız.

```
Mergesort3(A[0..n-1]):
if n ≤ 1, then return (A[0..n-1]).
Let k = ⌊n/3⌋ and m = ⌊2n/3⌋.
Return Merge3(Mergesort3(A[0..k-1]), Mergesort3(A[k..m-1]), Mergesort3(A[m..n-1]), A[0..n-1]).
```

```
Merge3(B,C,D,X):
Merge(B,C,E); Merge(E,D,X).
```

b) En kötü durum senaryosunda, her biri  $(n/3)$  uzunluğundaki üç sıralı listeyi birleştirmek için gerekli anahtar karşılaştırma sayısını belirleyiniz. ( $O(\cdot)$  notasyonu kullanınız.)

$$n/3 + n/3 - 1 = 2n/3 - 1 \text{ for Merge}(B,C,E);$$

$$2n/3 + n/3 - 1 = n - 1 \text{ for Merge}(E,D,X);$$

$$\text{Total: } 5n/3 - 2 \in O(n)$$

c) Üçlü birleştirme sıralama algoritmasına ilişkin en kötü durum çalışma zamanına ilişkin özyineleme bağıntısını yazıp çözünüz.

$$T(n) = 3T(n/3) + O(n)$$

Master teoremine dayalı olarak;

- Bir özyineleme ilişkisi için

$$T(n) = aT(n/b) + f(n), \quad f(n) \in \Theta(n^d), \quad d \geq 0$$

**Master Theorem** If  $f(n) \in \Theta(n^d)$  where  $d \geq 0$  in recurrence

$$T(n) \in \begin{cases} \Theta(n^d) & \text{if } a < b^d, \\ \Theta(n^d \log n) & \text{if } a = b^d, \\ \Theta(n^{\log_b a}) & \text{if } a > b^d. \end{cases}$$

$$T(n) = O(n \log n)$$

**Soru#4 (20 puan)**

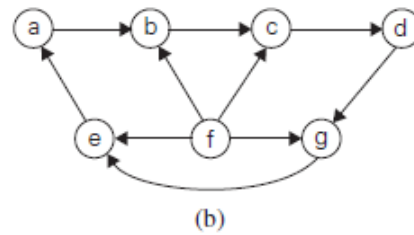
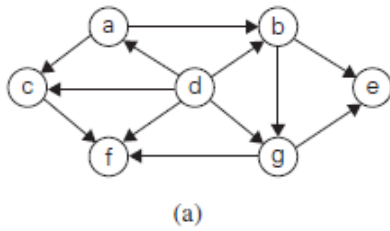
$A[0..n-2]$  dizisi, 1'den  $n$ 'ye kadar  $n-1$  tamsayıyı artan sırada içermektedir. Ancak, belirtilen değer aralığındaki bir tamsayı eksiktir. **Dizideki eksik olan sayıyı belirlemek için bir algoritma tasarlayıp algoritmanızın sözde kodunu yazınız.**

Problem azalt ve fethet yaklaşımına dayalı bir algoritma tasarlanarak çözülebilir. Eğer sıralı listenin orta elemanı  $A[m]$  içeriğini  $m+1$  ile karşılaştırırsak, algoritmanın eksik değeri aramaya nereden devam etmesi gerektiğini belirleyebiliriz. Eğer  $A[m]=m+1$  ise eksik olan değer  $m+1$ 'den büyüktür. Dolayısı ile dizinin sağ tarafında aramaya devam edilebilir. Aksi takdirde, aramaya sol tarafta devam edilmelidir.

```

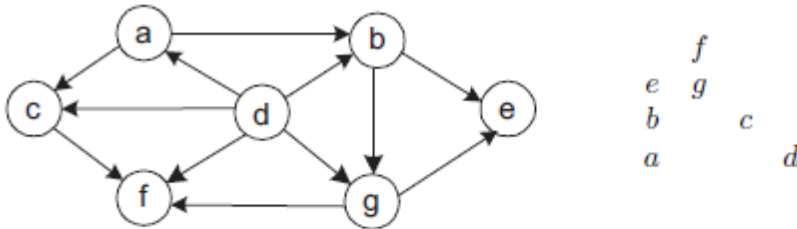
Algorithm MissingNumber( $A[0..n-2]$ )
//Input: An increasing array of  $n-1$  integers in the range from 1 to  $n$ 
//Output: An integer from 1 to  $n$  that is not in the array
 $l \leftarrow 0$ ;  $r \leftarrow n-2$ 
while  $l < r$  do
     $m \leftarrow \lfloor (l+r)/2 \rfloor$ 
    if  $A[m] = m+1$ 
         $l \leftarrow m+1$ 
    else  $r \leftarrow m-1$ 
if  $A[l] = l+1$  return  $l+2$ 
else return  $l+1$ 

```

**Soru#5 (20 puan)**

Yukarıda verilen yönlü çizgeler üzerinde, **derinlik öncelikli arama** algoritmasını uygulayarak topolojik sıralama problemini çözünüz.

a. The digraph and the stack of its DFS traversal that starts at vertex  $a$  are given below:



The vertices are popped off the stack in the following order:

$e f g b c a d.$

The topological sorting order obtained by reversing the list above is

$d a c b g f e.$

b. The digraph below is not a dag. Its DFS traversal that starts at  $a$  encounters a back edge from  $e$  to  $a$ :

