Levitin, A. (2008). *Introduction To Design And Analysis Of Algorithms,* Pearson Education USA, 2.Edition.

Algoritmalar

#1.A.1

Aşağıdakilerden hangisi daha büyük bir zaman karmaşıklığını ifade eder?

A) O(n!)

B) O(n)

C) O(nlogn)

D) $O(n^2)$

E) O(logn)

Cevap: A) O(n!)

İki algoritmanın zaman karmaşıklıkları incelenirken girdi boyutu 'n' sonsuza giderken algoritmanın çalışma zamanının nasıl tepki verdiği gözlemlenir. Seçeneklerdeki fonsiyonlara bakıldığında n sonsuza giderken en hızlı artacak fonksiyon n! fonksiyonudur.

#2.A.3 Aşağıda verilen kod bloğunun zaman karmaşıklığı $BigTheta(\Theta)$notasyonu türünden nedir?

```
ALGORTIHM (n, m)
  temp ← 0
  for i ← 0 to n+1 do
    temp ←temp + m
  end for
  for j ← 0 to 5 do
    temp ← temp + n
  end for
return temp
```

A) $\Theta(n)$

B) $\Theta(nlogn)$

C) $\Theta(n + m)$

D) $\Theta(logn)$

E) $\Theta(n^2)$

Cevap: A) $\Theta(n)$

Bir algoritmanın zaman karmaşıklığı hesaplanırken öncelikle algoritmanın çalışma zamanına etki edecek parametreler yani girdi boyutu belirlenir. Sorudaki algoritmanın 'n' ve 'm' olmak üzere iki adet parametresi vardır. Bu parametrelerden 'm' çalışma zamanına etki etmez sadece 'n' çalışma zamanına etki eder. Kod bloğunun içerisinde iki adet for döngüsü var. Bir tanesi 0'dan (n+1)'e kadar diğeri ise 0'dan 5'e kadar. Dolayısı ile algoritmanın çalışma zamanı girdi boyutu 'n' ile doğrusal bir ilişkisi vardır diyebiliriz ve cevabımız $\Theta(n)$ olur.

#3.A.3 Prim algoritması, verilen ağırlıklı ve yönsüz bir graftaki minimum kapsama ağacını çıkaran algoritmalardan bir tanesidir. Prim algoritması aşağıdaki algoritma tasarım tekniklerinden hangisini kullanır?

A) Açgözlü (hırslı) tasarım tekniği
B) Nesneye dayalı programlama tekniği
C) Azalt ve çöz tekniği
D) Böl ve yönet tekniği
E) Dinamik programlama

Cevap: A) Açgözlü (hırslı) tasarım tekniği

Hırslı tasarım tekniği: Programcı problem kısıtlarına ek olarak çözüm uzayının tamamını dolaşmamak ve daha hızlı çözüm bulabilmek için kendi kısıtlarını da orjinal probleme ekler. Bu kısıtlar iteratif olarak problem çözümüne ulaşmak için uygulanır ve her iterasyonda kısıtlara uyan en ideal sonuç geri dönmeksizin elde edilir. Bu sebepten dolayı bu algoritmalar hırslı (açgözlü) algoritmalar olarak isimlendirilir ve her problem için global optimumu garanti etmezler. Yani bazı problem tiplerinde local optimuma takılabilirler. Prim algoritması da bahsedilen özellikleri barındıran açgözlü teknik kullanılarak tasarlanmış algoritmalardan bir tanesidir.

#4.A.1 Hash fonksiyonlarının genel özellikleri gözönünde bulundurulduğunda aşağıdaki bilgilerden hangisi yanlıştır?

A) Arama işleminin zaman karmaşıklığını artırır.

B) Mümkün mertebe çakışmayı önlemeye çalışırlar.
C) Tersinir çalışmazlar. Çıktıdan girdi elde edilemez.
D) Çıktıları tahmin edilebilir değildir.
E) Girdi olarak aldıkları aynı veriyi çıktı uzayında her zaman aynı yere atarlar.

Cevap: A) Arama işleminin zaman karmaşıklığını artırır.

Hash fonksiyonları günümüzde siber güvenlik gibi alanlarda farklı amaçlar için kullanılsalar da ilk ortaya çıktıklarında veri yapısı olarak kullanılmaktaydılar. Bilgisayarların sıkça yaptığı işlemlerden bir tanesi arama işlemidir. Hash fonksiyonları veri yapısı olarak düşünüldüğünde faydası arama işlemlerinin daha hızlı yapılabilmesi için ilgili kayıtları matematiksel bir model yardımı ile kodlayarak saklamasıdır. Yani bir bakıma dizinleme işlemi yaparlar. Kayıt tekrar istendiğinde kaydı ilgili dizine giderek daha hızlı bulurlar dolayısı ile Hash fonksiyonları arama zaman karmaşıklığını azaltırlar.

#5.A.5

Aşağıda Mystery ismi ile verilen fonksiyon ne yapmaktadır?

ALGORITHM Mystery(A[0 … n-1)

```
for i ← 0 to n-2 do
    for j ←i+1 to n-1 do
        if A[i] = A[j]
            return false
        end if
    end for
end for
return true
```

A) Parametre olarak gelen dizinin eşsiz elemanlardan oluşup oluşmadığını bulur.
B) Parametre olarak gelen dizideki en büyük sayıyı bulur.
C) Parametre olarak gelen dizideki en küçük sayıyı bulur.
D) Parametre olarak gelen dizinin medyan değerini bulur.
E) Parametre olarak gelen dizideki elemanların aritmetik ortalamasını bulur.

Cevap: A) Parametre olarak gelen dizinin eşsiz elemanlardan oluşup oluşmadığını bulur.

Soruda verilen kod bloğu kendisine parameter olarak gelen dizideki elemanları birbirleri ile kıyaslayarak benzer olup olmadıklarına bakıyor. Eğer dizi içerisinde aynı elemanlar var ise fonksiyon ‘false’ değeri ile geri dönüyor. Eğer dizi benzersiz (eşsiz) elemanlardan oluşmuş ise if

bloğunun içerisine hiç girmiyor ve dizi ‘true’ değeri ile geri dönüyor. Dolayısı ile soruda verilen kod bloğu bir dizinin eşsiz elemanlardan oluşup oluşmadığını bulur.