

BİL 206 – Algoritma Analizi ve Tasarımı

Ödev - 01

Teslim Tarihi 5 Nisan – 23:59

Ödev Kuralları:

- Ödev kağıdınızın en üstüne adınız-soyadınızı ve öğrenci numaranızı açık bir şekilde yazınız.
- Sadece el yazısı ile hazırlanmış ödevler değerlendirilecektir. Ödev temiz bir el yazısı ile okunur bir şekilde hazırlanmalı, her bir soru için ulaşılan nihai sonuç dikdörtgen içine alınmalıdır.
- Ödev tesliminde taranmış ve tek bir pdf'e dönüştürülmüş bir dosyayı yükleyeceksiniz. Pdf ismi olarak sadece öğrenci numaranızı kullanabilirsiniz.
- Bu ödevi bireysel olarak tamamlayacaksınız. Cevapları birbirinize açıkça söylemeniz, ödevi kopyalamayınız, ödevinizin kopyalanmasına izin vermeniz durumunda, kopya ödev tespit edilirse ödevinizden sıfır puan alırsınız.
- Son teslim tarihinden sonra gönderilen ödevler puanlanmayacaktır.
- Verilen ödevlerin toplam geçme notuna etkisi en az %20 en fazla %25 olacaktır. (Bu dönem içerisinde toplamda üç ödev verilmesi planlanmaktadır.)
- Ödev soru kağıdınızın bir kopyasını muhafaza ediniz. Ödev cevapları tamamı ile yayınlanmayacaktır. Sadece nihai ulaşılan sonuca ait kısa cevaplar açıklanacaktır. Ödev değerlendirmesinden sonra nihai sonuçlara ait cevaplarınızı kontrol ettiğinizde, cevabınızın doğruluğu ile ilgili kontrolü yapabilirsiniz. Ödevler ile ilgili dersin asistanı Fatih Kayrancı hocanız ile iletişime geçiniz (fatih.kayranci@medeniye.edu.tr).

1) Aşağıda verilen fonksiyon için:

```
Function Func1 (n)
// Input: n is an integer
// Output: cnt is an integer
{
    cnt = 0
    for i=1 to sqrt(n)
        for j=1 to i
            for k=1 to j*j
                cnt = cnt + 1
    return cnt
}
```

- a) Yukarıdaki algoritmada temel işlemi (operasyonu) nedir?
- b) Temel işlem kaç kez çalıştırılır?
- c) Bu algoritmanın zaman karmaşıklığını (time complexity) O (büyük O), Ω (büyük omega) ve Θ (teta) sonuçur gösterimleriyle (asymptotic notation) ifade ediniz (b. şıktaki hesaplamayı temel alınız).
- d) Algoritma için o (küçük o) ve ω (small omega) sonuçur gösterimi için uygun olan zaman karmaşıklığı yazınız (Her biri için iki farklı karmaşıklık yazınız).
- e) A bilgisayarında algoritma problem büyüklüğü 10 için çalışması 8 saniye sürmektedir.

Problem büyüklüğü 160 olduğunda algoritmanın A bilgisayarında çalışma süresi ne olacaktır?

f) Algoritmanın yer karmaşıklığı (space complexity) nedir?

2) Aşağıda verilen fonksiyon için:

```
void replace (int* x, int* y)
{ int z = *y;
  *y = *x;
  *x = z; }

void Func2(int A[ ], int n)
{ int ind = 0;
  while (ind < n)
  {
    if (ind == 0)
      ind++;
    if (A[ind] >= A[ind - 1])
      ind++;
    else
    {
      replace(A[ind], A[ind - 1]);
      ind--;
    }
  } }
```

- Yukarıdaki algortmada verilen bir dizi için sonuç olarak ne döndürmektedir? Bir örnek üzerinden açıklayınız.
- Temel işlem kaç kez çalıştırılır?
- Bu algoritmanın zaman karmaşıklığını (time complexity) O (büyük O), Ω (büyük omega) ve Θ (teta) sonuçur gösterimleriyle (asymptotic notation) aşağıdaki durumlar için gösteriniz.
 - En kötü durum analizi (worst case)
 - En iyi durum analizi (best case)
 - Ortalama durum analizi (average case)
- Yukarıdaki her durumu temel alarak algoritmanın o (küçük o) ve ω (small omega) sonuçur gösterimi için uygun olan zaman karmaşıklığı yazınız (Her biri için iki farklı karmaşıklık yazınız).
- Algoritmanın yer karmaşıklığı (space complexity) nedir?

3) Aşağıda verilen fonksiyon için:

```
Function Func3 (A[1...n])
// Input : an array of integers
// Output : an integer
{
  if (n <= 1)
    return 4;
  else
    return Func3(A[1...n/3]) + Func3(A[2n/3...n]);
}
```

- a) Yukarıdaki algoritmada temel işlemi (operasyonu) nedir?
- b) Fonksiyon içerisinde temel operasyonun kaç kere yapıldığını özyinelemeli fonksiyon (recurrence relation) olarak yazınız.
- c) Özyinelemeli fonksiyonu çözerek fonksiyon içerisinde temel operasyonun kaç kere yapıldığını hesaplayınız.
- d) Algoritmanın karmaşıklığını en uygun sonuçur gösterimi kullanarak veriniz?
- 4) Aşağıda verilen her bir durum için $f \in O(g)$ veya $f \in \Omega(g)$ veya her ikisini birden (bu durumda $f \in \Theta(g)$) olup olmadığını ispatlayınız. İspatlamak için hem limit hem de tanım gereği kanıt (proof by definition) yöntemlerini kullanınız.

	f(n)	g(n)
a)	$n^{1.01}$	$n \log^2 n$
b)	$n!$	$(n+1)^n$
c)	\sqrt{n}	$\log\left(\frac{2^{n+1}}{n}\right)$
d)	$\sum_{i=1}^n i^k$	n^{k+1}
e)	$32^{\log n}$	n^6

- 5) Aşağıda verilen her bir yinelemeli bağıntısını master teorem, yerine koyma (substitution), yinemeli ağaç (recursion tree) yöntemleri ile çözünüz.

Her bir şık için yukarıda listelenen yöntemlerden birini kullanmanız yeterlidir. Fakat, her çözüm farklı bir yöntem ile gerçekleştirilmelidir.

i.	$T(n) = T(n-1) + 6 * T(n-2) , T(0) = 3, T(1) = 6$
ii.	$T(n) = 2T(n/4) + \sqrt{n} , T(1) = 1$
iii.	$T(n) = 4T(n-1) + 3^n , T(1) = 1$