

3. (20p) Aşağıda verilen f karmaşıklık fonksiyonlarını en kötü zaman tüketimi açısından küçükten büyüğe ($f=O(g)$) doğru açıklayarak sıralayınız. Çözüm adımları açıklanmayan cevaplar geçersiz sayılacaktır.

$$2^n n^3, n!, 2^{n^2}, n^{\sqrt{n}}, (\log n)^n.$$

2. (15p) Zaman karmaşıklığı $g(n)=4n^4$ olan bir algoritma ile işlemci hızı 2 GHz olan bilgisayarda 23 saatte kaç veri işlenebileceğini açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır. Düz metin alanında cevaplanması istenilen sorularda verilen linkler dikkate alınmayacaktır.

Soru 4

Henüz yanıtlanmadı 25.00 üzerinden alınan puan 4 Soruyu işaretle

4. (25p) Aşağıdaki algoritmanın ne iş yaptığını, en iyi durum analizi, en kötü durum analizi ve ortalama durum analizi ile karmaşıklığını bulunuz. Çözüm adımları açıklanmayan cevaplar geçersiz sayılacaktır.

```
Program(L)
  k=length(L)-1
  t1=L[k]
  k=k-1
  while k>=0
    if L[k]>t1
      t1=L[k]
    k=k-1
  return t1
```

4. (20p) Verilen bir dizideki en küçük sayıyı bulan böl-yönet algoritma stratejisine göre yazılmış $O(\log n)$ karmaşıklığında bir algoritmayı açıklayarak yazınız ve tekrarlı bağıntısını veriniz. Örneğin verilen dizi [6 4 2 4 7 9] ise 2 döndürecek. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır. Düz metin alanında cevaplanması istenilen sorularda verilen linkler dikkate alınmayacaktır.



3. (15p) Aşağıdaki algoritmalar için en kötü çalışma zamanını tanımlayan bir recurrence (tekrarlı bağıntı) vererek ve algoritmik karmaşıklıklarını açıklayarak bulunuz.

- a) Binary search
b) Insertion sort
c) Merge sort

Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır. Düz metin alanında cevaplanması istenilen sorularda verilen linkler dikkate alınmayacaktır.



1. (30p) a) Aşağıdaki program parçalarının her biri için Big-O gösterimi ile çalışma zamanının analizini veriniz.

b) Eğer $n=100$ için (b) programının çalışması 10 ms alıyorsa, $n=400$ için çalışması ne kadar süreceğini bulunuz.

c) Eğer $n=100$ için (a) programının çalışması 10 ms alıyorsa, 40 ms'de çözebileceği problem boyutunu bulunuz.

Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır. Düz metin alanında cevaplanması istenilen sorularda verilen linkler dikkate alınmayacaktır.

(a)

```
sum = 0;
for(i=0; i<sqrt(n)/2; i++)
    sum++;
for(j=0; j<sqrt(n)/4; j++)
    sum++;
for(k=0; k<8+j; k++)
    sum++;
```

(b)

```
sum = 0;
for(i=0; i<sqrt(n)/2; i++)
```

(c)

```
sum = 0;
for(i=1; i<2*n; i++)
    for(j=1; j<i*i; j++)
        for(k=1; k<j; k++)
            if (j % i == 1)
                sum++;
```

(d)

```
sum = 0;
for(i=1; i<2*n; i++)
    for(j=1; j<i*i; j++)
        for(k=1; k<i*k; k++)
```

boyutunu bulunuz.

Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır. Düz metin alanında cevaplanması istenilen sorularda verilen linkler dikkate alınmayacaktır.

(a)

```
sum = 0;
for(i=0; i<sqrt(n)/2; i++)
    sum++;
for(j=0; j<sqrt(n)/4; j++)
    sum++;
for(k=0; k<8+j; k++)
    sum++;
```

(b)

```
sum = 0;
for(i=0; i<sqrt(n)/2; i++)
    for(j=i; j<8+i; j++)
        for(k=j; k<8+j; k++)
            sum++;
```

(c)

```
sum = 0;
for(i=1; i<2*n; i++)
    for(j=1; j<i*i; j++)
        for(k=1; k<j; k++)
            if (j % i == 1)
                sum++;
```

(d)

```
sum = 0;
for(i=1; i<2*n; i++)
    for(j=1; j<i*i; j++)
        for(k=1; k<j; k++)
            if (j % i)
                sum++;
```

4. (25p) Uzunlukları aynı olan A ve B dizilerinin birebir aynı olup olmadığını bulan böl-yönet stratejisine göre bir algoritma yazınız. Bu algoritmanın tekrarlı bağıntısını yazınız. Sonrasında bu tekrarlı bağıntıya karşılık gelen algoritmik karmaşıklığı açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.

2. (15p) Zaman karmaşıklığı $g(n)=5n^3$ olan bir algoritma ile işlemci hızı 5 GHz olan bilgisayarda 13 dakikada kaç veri işlenebileceğini açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.

3. (30p) Aşağıda verilen algoritmaların karmaşıklıklarını açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.

<pre>public static int f6(int N) { if (N == 0) return 1; return f6(N-1) + f6(N-1) + f6(N-1); }</pre>	<pre>int x = 0, i, j; for(i = 1; i <= N; i++) for(j = 1; j <= N+R; j+=i) x += j;</pre>
a	b
<pre>public static int f7(int N) { int x = 0; while (N > 0) { x++; N = N / 2; } return x; }</pre>	<pre>public static int f2(int N) { int x = 1; while(x < N) x = x * 2; return x; }</pre>
c	d
<pre>int x = 0, i; for(i = 0; i < N; i++) x += f2(N);</pre>	<pre>public static int f3(int N) { if (N == 0) return 1; int x = 0; for (int i = 0; i < N; i++) x += f3(N-1); return x; }</pre>
e	f

<pre>public static int f2(int N) { int x = 1; while(x < N) x = x * 2; return x; }</pre>	<pre>int x = 0, i; for(i = 0; i < N; i++) x += f2(N);</pre>
a	b
<pre>int x = 0, i, j; for(i = 1; i <= N; i++) for(j = 1; j <= N+R; j+=i) x += j;</pre>	<pre>public static int f3(int N) { if (N == 0) return 1; int x = 0; for (int i = 0; i < N; i++) x += f3(N-1); return x; }</pre>
c	d
<pre>public static int f6(int N) { if (N == 0) return 1; return f6(N-1) + f6(N-1) + f6(N-1); }</pre>	<pre>public static int f7(int N) { int x = 0; while (N > 0) { x++; N = N / 2; } return x; }</pre>
e	f

Soru 3



Henüz yanıtlanmadı 20.00 üzerinden alınan puan Soruyu işaretle

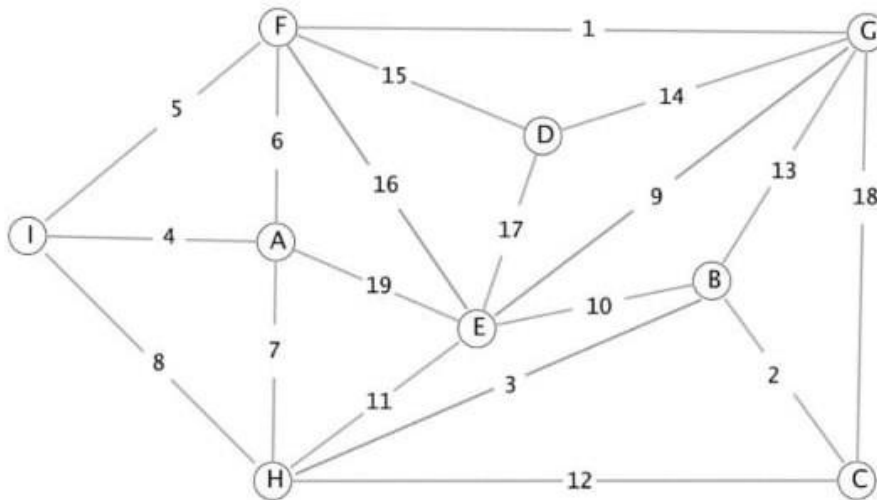
3. (20p) Aşağıda giriş boyutu N olan iki farklı program için farklı giriş boyutlarına karşılık algoritmanın tükettiği süreler verilmiştir. Buna göre;
- a) (a) şıkkındaki algoritmanın karmaşıklığını açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.
- b) (b) şıkkındaki algoritmanın giriş boyutu $N=200.000$ olursa algoritmanın tüketeceği süreyi açıklayarak bulunuz.

N	time	N	time
125	0.03 sec	5,000	0.2 seconds
1,000	1.00 sec	10,000	1.2 seconds
8,000	32.00 sec	20,000	3.9 seconds
64,000	1,024.00 sec	40,000	16.0 seconds
512,000	32,768.00 sec	80,000	63.9 seconds
a		b	

Soru 2

Henüz yanıtlanmadı 20.00 üzerinden alınan puan Soruyu işaretle

2. (20p) Aşağıda verilen ağırlıklı grafa göre Minimum Spanning Tree'yi bulmak için Prim ve Kruskal algoritmalarını kullandığınızda elde edilecek ağacın ağırlıklarını algoritmaları uygularken oluşacak sıraya göre açıklayarak yazınız. Kruskal algoritması için 1 ağırlıklı ayrıttan, prim algoritması için 4 ağırlıklı ayrıttan başlayınız. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.



Soru 4

Henüz yanıtlanmadı 30.00 üzerinden alınan puan  Soruyu işaretle

4. (30p) Uzunlukları aynı olan A ve B dizilerinin birebir aynı olup olmadığını bulan böl-yönet stratejisine göre bir algoritma yazınız. Bu algoritmanın tekrarlı bağıntısını yazınız. Sonrasında bu tekrarlı bağıntıya karşılık gelen algoritmik karmaşıklığı açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.

1. (30p) Aşağıda verilen algoritmaların karmaşıklıklarını açıklayarak bulunuz. Çözüm adımlarının açıklamaları yazılmadan verilen cevaplar geçersiz sayılacaktır.

<pre>public static int f1 (int n) { int x = 0; for (int i = 0; i < n; i++) x++; return x; }</pre>	<pre>public static int f2(int n) { int x = 0; for (int i = 0; i < n; i++) for (int j = 0; j < i*i; j++) x++; return x; }</pre>
<u>a</u>	<u>b</u>
<pre>public static int f3 (int n) { if (n <= 1) return 1; return f3(n-1) + f3(n-1) }</pre>	<pre>public static int f4 (int n) { if (n <= 1) return 1; return f4(n/2) + f4(n/2); }</pre>
<u>c</u>	<u>d</u>
<pre>public static int f5 (int n) { if (n <= 1) return 1; return f1(n) + f5(n/2) + f5(n/2); }</pre>	<pre>public static void f6(int n) { // 1<i is the same as 2ⁱ. // Ignore integer overflow. // 1<i takes constant time. for (int i = 0; i < n; i=1<i); }</pre>
<u>e</u>	<u>f</u>

1. (25p) n elemanlı bir diziyi sıralamak için merge sort algoritması kullanılacaktır. $n=13$ için bu algoritmanın recursion ağacı çizilirse; Recursion ağacındaki seviye sayısı nedir. En kötü durumda her bir seviyedeki karşılaştırma sayısı nedir. İhtiyaç duyulan toplam karşılaştırma sayısı nedir. n 2'nin kuvveti olmak üzere n sayıdaki eleman için seviye sayısı, her seviyedeki karşılaştırma sayısı ve toplam karşılaştırma sayısı nedir. Ayrıntılı olarak açıklayınız. Çözüm adımları açıklanmayan cevaplar geçersiz sayılacaktır.
2. (10p) a) Zaman karmaşıklığı $g(n)=3.5n^3$ olan bir algoritma ile işlemci hızı 3.5 GHz olan bilgisayarda 3.5 dakikada kaç veri işlenebilir. Çözüm adımları açıklanmayan cevaplar geçersiz sayılacaktır.
- (20p) b) Aşağıdaki program parçalarının her biri için Big-O notasyonu ile çalışma zamanı analizini açıklayarak veriniz. Çözüm adımları açıklanmayan cevaplar geçersiz sayılacaktır.

a) sum=0; for(i=0;i<n;i++) sum++; for(j=0;j<n;j++) sum++;	b) sum=0; for(i=0;i<n/2;i++) for(j=i;j<n/4;j++) sum++;
c) sum=0; for(i=1;i<n;i++) for(j=1;j<i*i;j++) for(k=1;k<j;k++) if(j%i==0) sum++;	d) sum=0; for(i=0;i<n;i++) for(j=0;j<i*i;j++) for(k=0;k<=j;k++) sum++;