

Algoritma Analizi

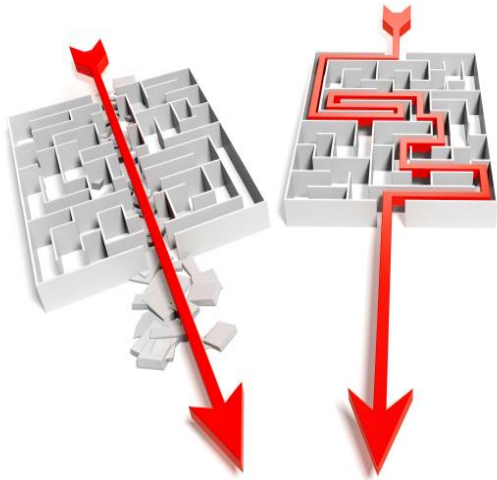


Suham SAHIN
Onur GÖK

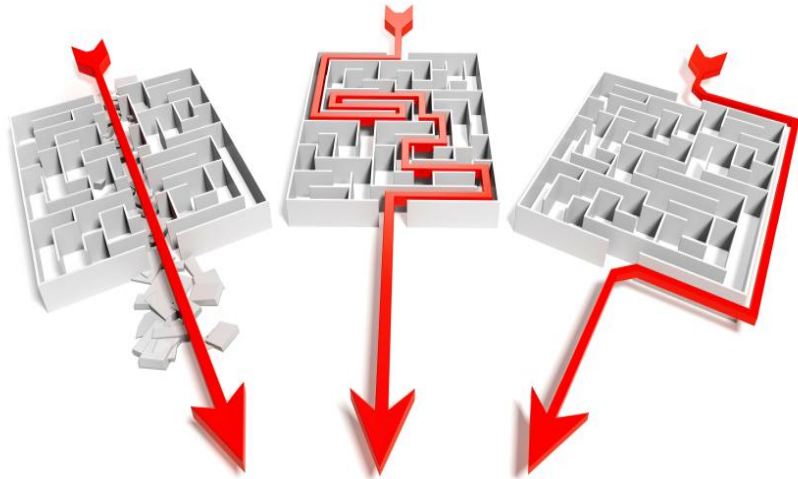
Giris



Giris



Giris



Verimlilik



Karşılaştırma

Giris



Input

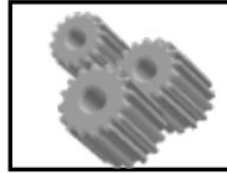
Giris



Input

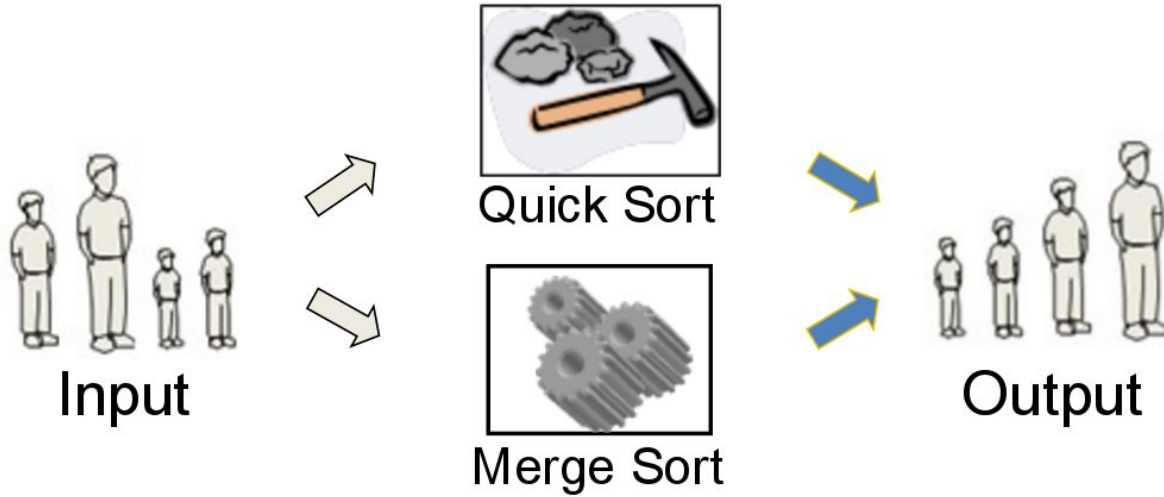


Quick Sort

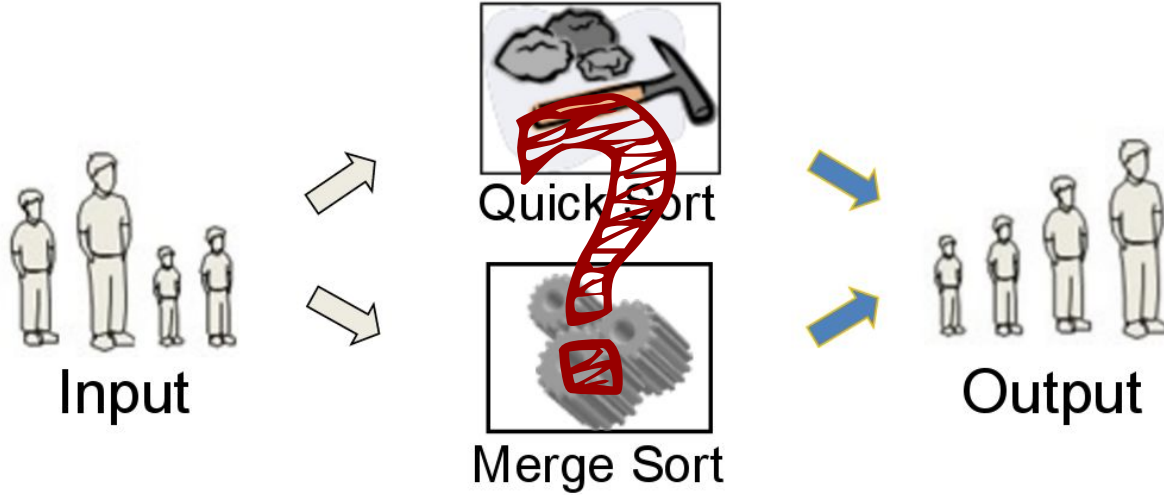


Merge Sort

Giris



Giris

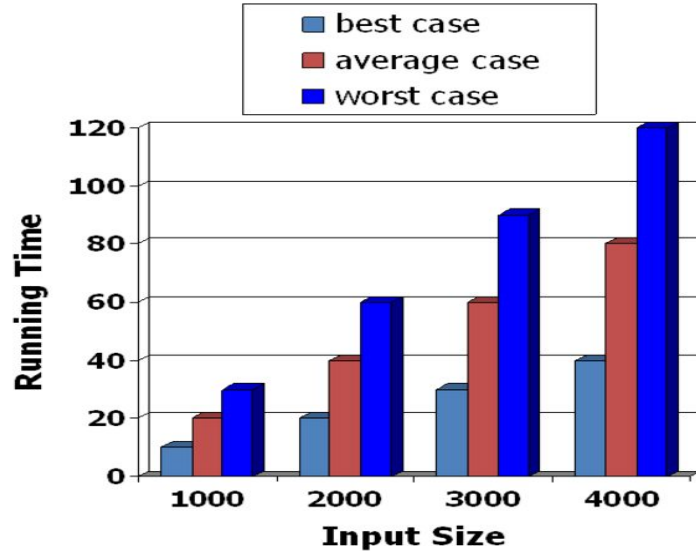


Hangisi daha iyi?

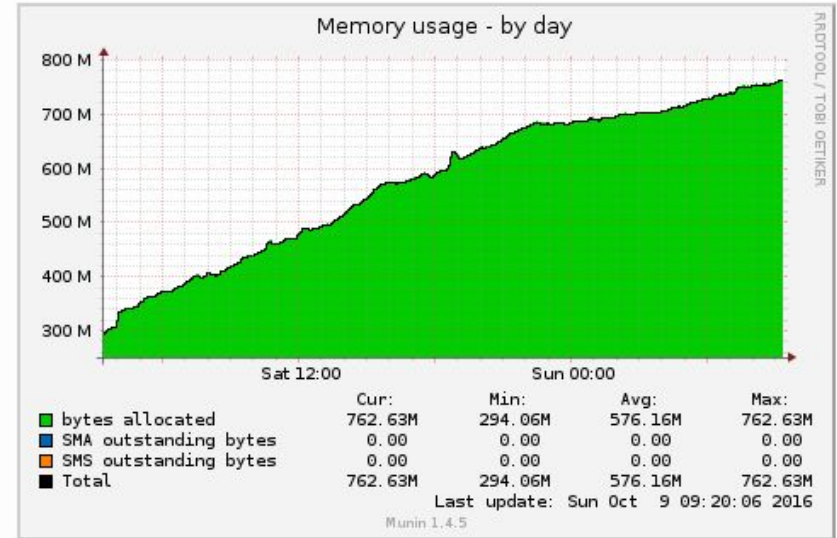
Hangi kriter gere göre?

Verimin ölçülmesi

Çalışma Zamanı



Bellek kullanımı



Verimin ölçülmesi

insan maliyeti

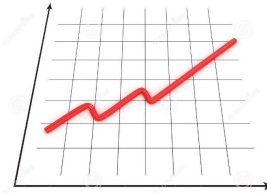


Veriminin hesaplanması

Çalışma Zamanı

Karşıklık

Parametre sayısı \times (artım)



Çalışma zamanı

Gerekli işlem miktarı



Verim hesaplama

Verim, bilgisayar ve gerçekteştirime bağı Matematiksel bir analiz

Kıyaslama



Algoritma Analizi

 Σ

`\sum`

 \cap

`\bigcap`

 \odot

`\bigodot`

 \prod

`\prod`

 \cup

`\bigcup`

 \otimes

`\bigotimes`

 \coprod

`\coprod`

 \sqcup

`\bigsqcup`

 \oplus

`\bigoplus`

 \int

`\int`

 \vee

`\bigvee`

 \biguplus

`\biguplus`

 \oint

`\oint`

 \wedge

`\bigwedge`

Algoritma Analizi

Matematiksel ifadesi nedir?

Temel hesap birimi kaç adet yapılmalı?

n
Parametre sayısı



Input



Algorithm



Output

$f(n)$
Verimlilik

n



$T(n)$ Çalışma süresi
 $S(n)$ Bellek gereksinimi

Çalışma Süresi Hesabı

Çalışma Süresi (Tn): Algoritmanın belirli bir isleme (Temel hesap birimi) kaç kez gereksinim duyduğunu gösteren bağıntıdır.

Temel hesap birimi:

- Programlama dilindeki deyimler
- Döngü sayısı
- Toplama işlem sayısı
- Dosyaya erişim sayısı
- Atama sayısı

IF-else { C O D E }

Loop

ÇSH: Aritmetik Ortalama

```
1  float bulOrta(int A[ ],int n){
2      float ortalama, toplam=0;
3      int k;
4      for (k=0; k<n;k++)
5          toplam +=A[k]; // döngü içinde gerçekleşen işlem
6      ortalama=toplam/n;
7      return ortalama;
8  }
```

ÇSH: Aritmetik Ortalama

	Temel Hesap Birimi	islem	tekrarı	Toplam
1	float bulOrta(int A[],int n){			
2	float ortalama, toplam=0;			
3	int k;			
4	for (k=0; k<n;k++)	1,1,1	1,(n+1),n	2n+2
5	toplam +=A[k];	1	n	n
6	ortalama=toplam/n;	1	1	1
7	return ortalama;	1	1	1
8	}			

$$T(n)=3n+4$$

ÇSH: En Küçük eleman

```
1  int bulEnkucuk(int A[], int n){  
2      int enkucuk;  
3      int k;  
4      enkucuk=A[0];  
5      for (k=0; k<n;k++)  
6          if(A[k]<enkucuk)  
7              enkucuk=A[k];  
8      return enkucuk;  
9  }
```

ÇSH: En Küçük eleman

	Temel Hesap Birimi	islem	tekrarı	Toplam
1	<code>int bulEnkucuk(int A[], int n){</code>			
2	<code>int enkucuk;</code>			
3	<code>int k;</code>			
4	<code>enkucuk=A[0];</code>	1	1	1
5	<code>for (k=0; k<n;k++)</code>	1,1,1	1,n,(n+1)	2n
6	<code>if(A[k]<enkucuk)</code>	1	n-1	n-1
7	<code>enkucuk=A[k];</code>	1	n-1	n-1
8	<code>return enkucuk;</code>	1	1	1
9	<code>}</code>			

$$T(n)=4n$$

ÇSH: Matris Toplama

```
1 void toplamMatris(int A[2][2], int B[2][2]){  
2     int C[2][2];  
3     int i,j;  
4     for(i=0;i<2;i++)  
5         for(j=0;j<2;j++)  
6             C[i][j]=A[i][j]+B[i][j];  
7 }
```

ÇSH: Matris Toplama

	Temel Hesap Birimi	islem	tekrarı	Toplam
1	<code>void toplamMatris(int A[2][2], int B[2][2]){</code>			
2	<code>int C[2][2];</code>			
3	<code>int i,j;</code>			
4	<code>for(i=0;i<2;i++)</code>	1,1,1	1,n,(n+1)	2n+2
5	<code>for(j=0;j<2;j++)</code>	1,1,1	n*(2m+2)	2nm+2n
6	<code>C[i][j]=A[i][j]+B[i][j];</code>	1	n*m	nm
7	<code>}</code>			

$$T(n)=3nm+4n+2$$

ÇSH: Faktöriyel Hesabı

```
1  int faktoriyel(int n){  
2      if(n<=1)  
3          return 1;  
4      else  
5          return (n*faktoriyel(n-1));  
6  }
```

ÇSH: Faktöriyel Hesabı

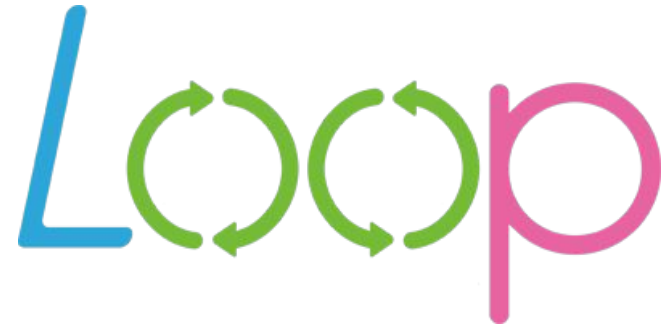
	Temel Hesap Birimi	islem	tekrarı	Toplam
1	<code>int faktoriyel(int n){</code>			
2	<code>if(n<=1)</code>	1	n	n
3	<code>return 1;</code>	1	1	1
4	<code>else</code>			
5	<code>return (n*faktoriyel(n-1));</code>	1	n-1	n-1
6	<code>}</code>			

$$T(n)=2^n$$

Karmasiklik

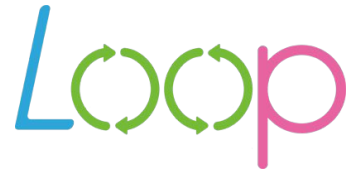
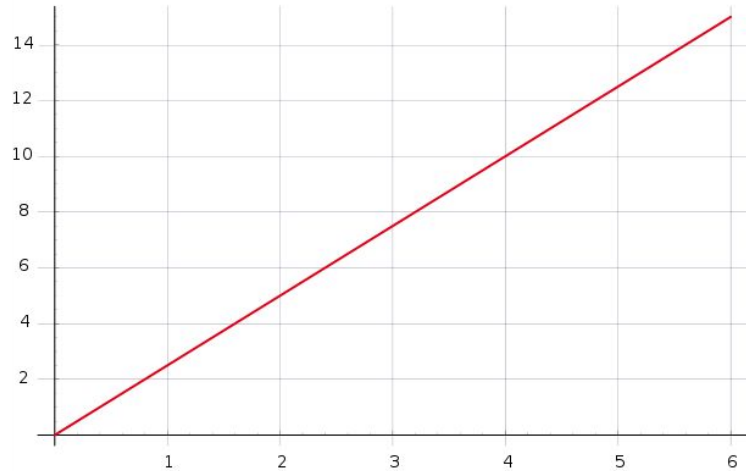
Karmasıklığı ifade edebilmek için matematiksel ifadeler kullanılmaktadır.

- ❖ Küçük-o (small-o)
- ❖ Büyük-O (big-o, veya big-oh diye de geçer)
- ❖ Teta (Theta Θ , sadece büyük tetadan bahsedebiliriz)
- ❖ Büyük omega (big- Ω)
- ❖ Küçük omega (small- ω)



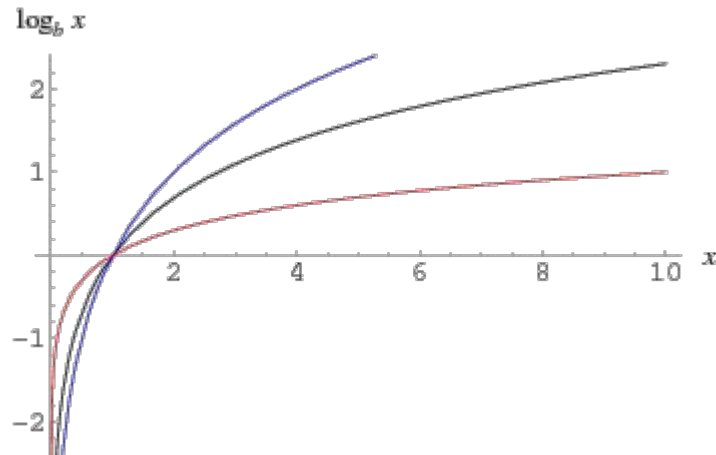
Linear Döngü

```
i = 1  
loop( i <= 1000 )  
    (loop body)  
    i = i + 1  
end loop
```



Logaritmik Döngü

```
i = 1  
loop( i <= 1000 )  
    (loop body)  
    i = i * 2  
end loop
```



Logaritmik Döngü

Çarpım Döngüsü

$i = 1$

loop($i \leq 1000$)

(loop body)

$i = i * 2$

end loop

iterasyon = $\log_2 i$

Çarpım	
iterasyon	i
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
çıkış	1024

Logaritmik Döngü

Bölüm Döngüsü

i = 1000

```
loop( i >= 1 )  
    (loop body)
```

i = i / 2

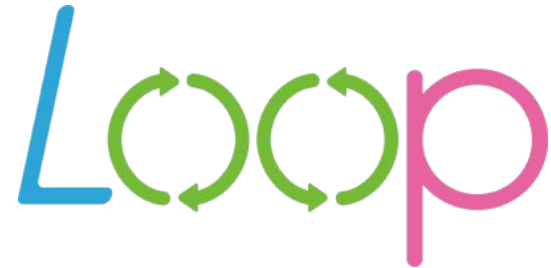
```
end loop
```

$$\text{iterasyon} = (\log_2 1000/i)$$

Bölüm	
iterasyon	i
0	1000
1	500
2	250
3	125
4	62
5	31
6	15
7	7
8	3
9	1
çıkış	0

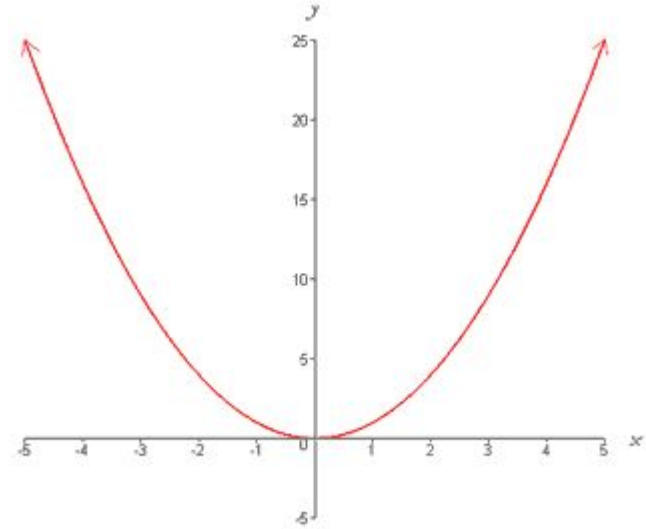
İç içe döngüler

- ❖ Karesel (Quadratic)
- ❖ Bağımlı Karesel (Dependent Quadratic)
- ❖ Lineer Logaritmik (Linear Logarithmic)



Karesel

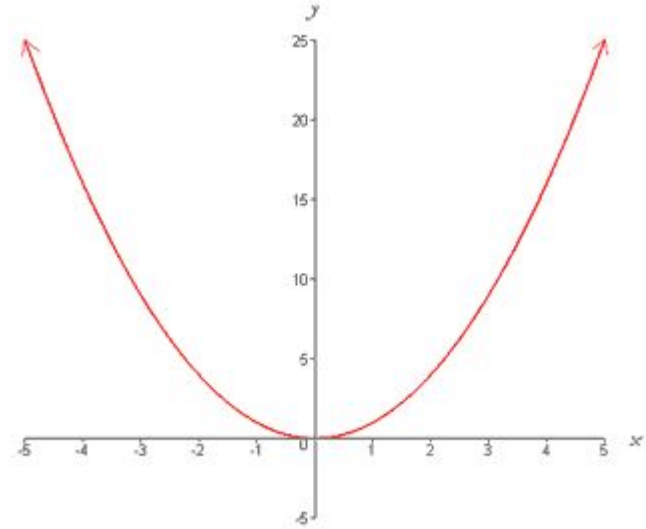
```
i = 1
loop( i <= n )
  j = 1
  loop( j <= n )
    (loop body)
    j = j + 1
  end loop
  i = i + 1
end loop
```



$$n^2 = n * n$$

Bagimli Karesel

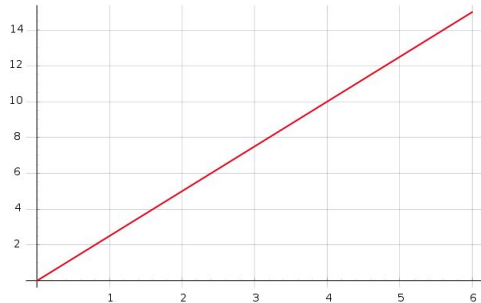
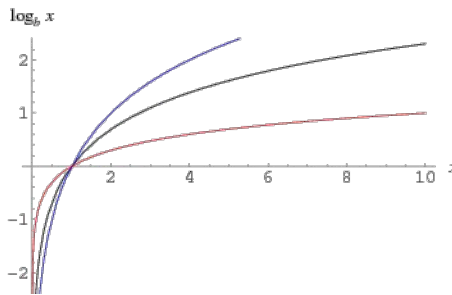
```
i = 1  
loop( i <= 10 )  
  j = 1  
  loop( j <= i )  
    (loop body)  
    j = j + 1  
  end loop  
  i = i + 1  
end loop
```



$$n^2 = n * n$$

Linear Logarithmik

```
i = 1
loop( i <= 10 )
  j = 1
  loop( j <= 10 )
    (loop body)
    j = j * 2
  end loop
  i = i + 1
end loop
```



Big O Gösterimi

$f(n)$ fonksiyonundan $O(n)$ değerini hesaplamak için aşağıdaki adımlar gerçekleştirilir:

- 1) Tüm katsayılar 1 yapılır
- 2) En büyük terim alınır, diğerleri gözardı edilir.



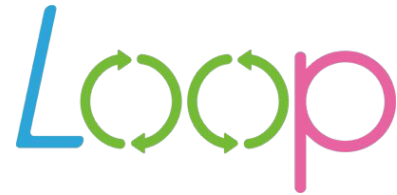
Big O Gösterimi

$$f(n) = \frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$$

$$n^2 + n$$

$$n^2$$

$$O(n^2)$$

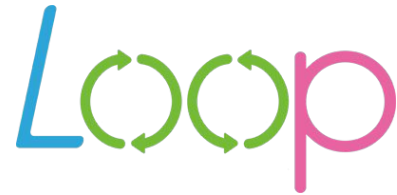


Big O Gösterimi

$$f(n) = 6n^4 \log n + 12n^3 + 2n^2 + n + 128$$

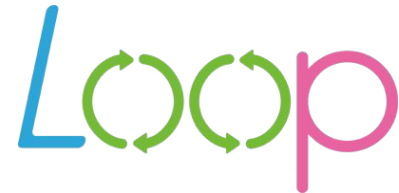
$$f(n) = n^4 \log n + n^3 + n^2 + n + 1$$

$$O(n^4 \log n)$$



Big O Gösterimi

Big O	Degisim sekli
$O(1)$	Sabit, degismiyor
$O(\log n)$	Logaritmik artıyor
$O(n)$	Dogrusal artıyor
$O(n \log n)$	Dogrusal çarpanlı logaritmik
$O(n^2)$	Karesel artıyor
$O(n^3)$	Kübik artıyor
$O(2^n)$	iki tabanında üssel artıyor
$O(10^n)$	On tabanında üssel artıyor
$O(n!)$	Faktöriyel olarak artıyor



Big O Gösterimi

Lineer Döngü

```
i = 1
loop( i <= 1000 )
    (loop body)
    i = i + 1
end loop
```

$$f(n)=n \Rightarrow O(n)$$

Logaritmik Döngü

//Çarpım Döngüsü

```
i = 1
loop( i < 1000 )
    (loop body)
    i = i * 2
end loop
```

$$2^i=n \Rightarrow f(n)=\log_2 n \Rightarrow O(\log n)$$

//Bölüm Döngüsü

```
i = 1000
loop( i >= 1 )
    (loop body)
    i = i / 2
end loop
```

Big O Gösterimi

// Karesel

```
i = 1
loop( i <= 10 )
  j = 1
  loop( j <= 10 )
    (loop body)
    j = j + 1
  end loop
  i = i + 1
end loop
```

$$f(n)=n*n \Rightarrow O(n^2)$$

// Bağımlı Karesel

```
i = 1
loop( i <= 10 )
  j = 1
  loop( j <= i )
    (loop body)
    j = j + 1
  end loop
  i = i + 1
end loop
```

$$f(n)=[n*(n+1)]/2 \Rightarrow O(n^2)$$

// Linear Logaritmik

```
i = 1
loop( i <= 10 )
  j = 1
  loop( j <= 10 )
    (loop body)
    j = j * 2
  end loop
  i = i + 1
end loop
```

$$f(n)=n*\log_2 n$$

$$O(n \log n):$$

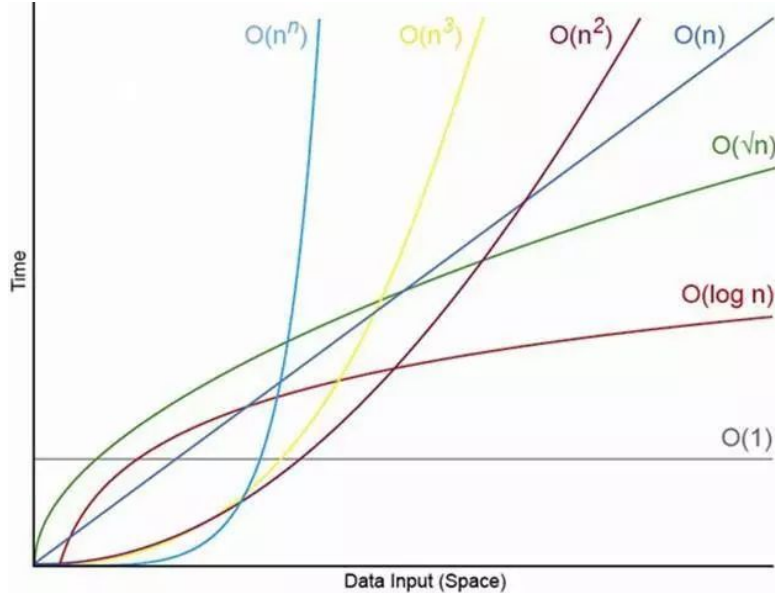
Büyük O Gösterimi

Big O : her şeyin ters gittiği durum



Büyük O Gösterimi

Algoritmanın büyüme katsayısı



Büyük O Gösterimi



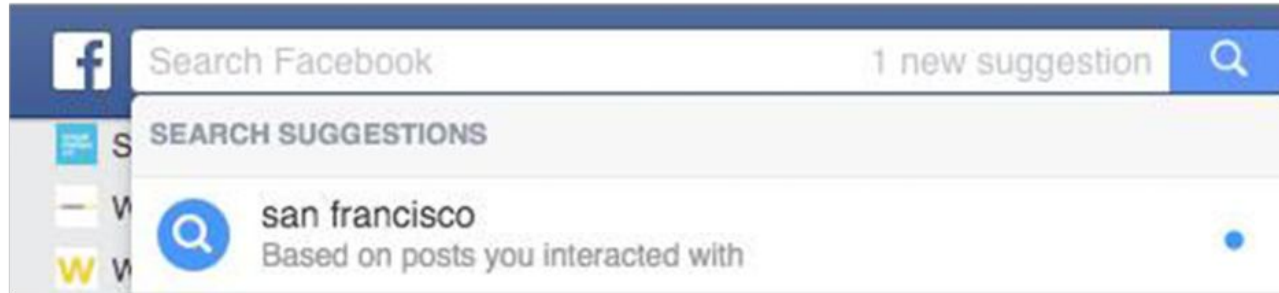
7.3
Billion

World Population

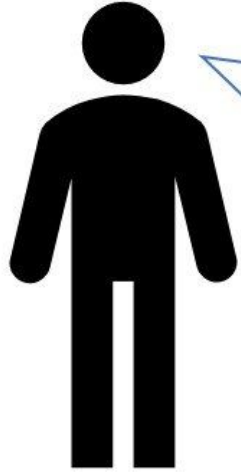


1.4
Billion

Facebook Users



Sabit Zaman $O(1)$



Sana pepsi
veriyorum



Sabit Zaman $O(1)$

```
char *icecekler[4]={"kola","sprite","pepsi","fanta"};  
char* icercek =icecekler[2];
```



Lineer Zaman $O(N)$



ilk iecek pepsi mi?
ikinci iecek pepsi mi?
nc iecek pepsi mi?
drdnc iecek pepsi mi?



Bulduk

Lineer Zaman $O(N)$

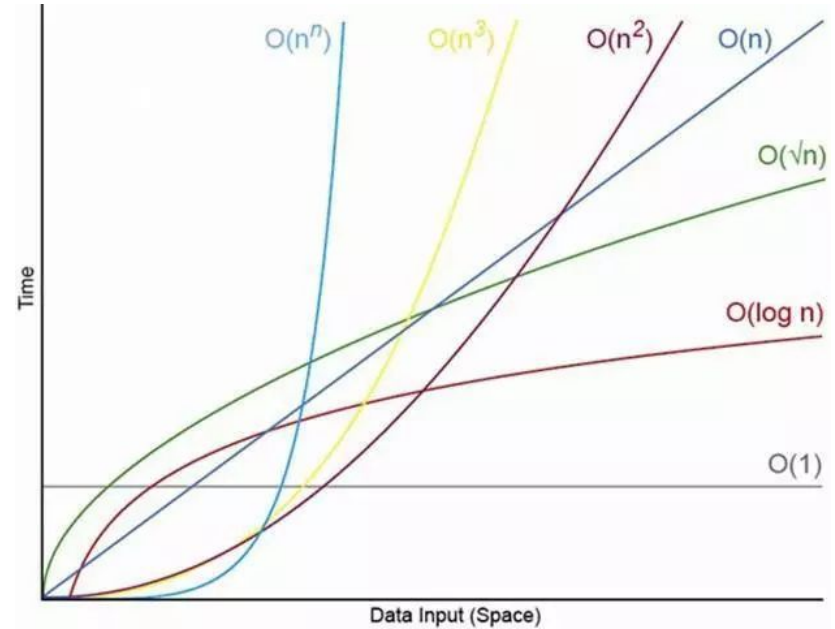
```
#include <stdio.h>
```

```
int icecekVarmi(char *istenilen, char *icecekler[]){
    char* icecek;
    int i=0;
    for(i=0;i<4;i++){
        icecek = icecekler[i];
        if(icecek == istenilen)
            return 1;
    }
    return 0;
}

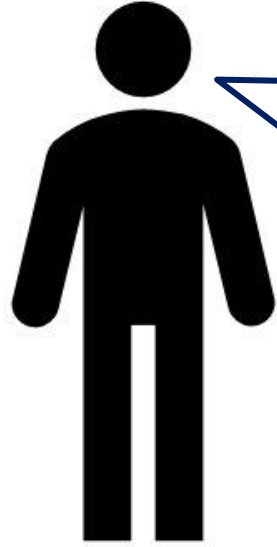
int main()
{
    char *icecekler[4]={"kola","sprite","pepsi","fanta"};
    int gelen = icecekVarmi("kola",icecekler);

    if(gelen) printf("buldum :");
    else printf("bulamadım :");

    return 0;
}
```



Logaritmik Zaman $O(\log N)$



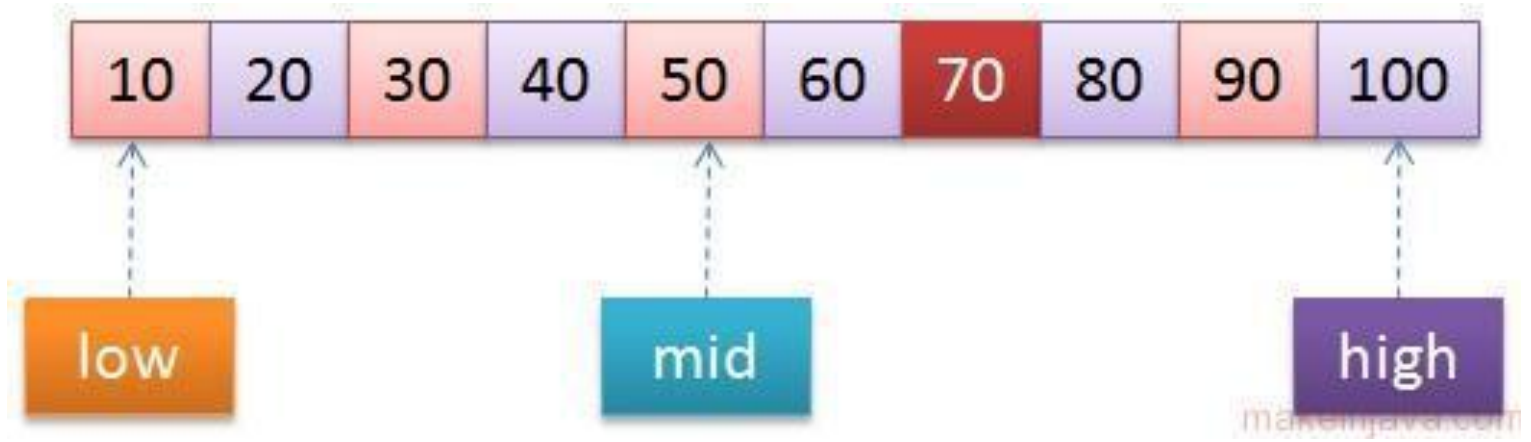
Sayı 1 ile 9 arasında tahmin et

Sayı 4? **Değil**

Sayı 4 den küçük? **Evet**

[~~1,2,3,4~~,5,6,7,8,9]

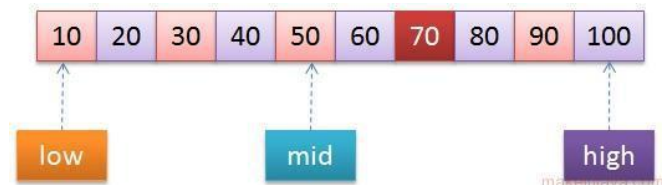
Logaritmik Zaman $O(\log N)$



[~~1,2,3,4~~,5,6,7,8,9]

Logaritmik Zaman $O(\log N)$

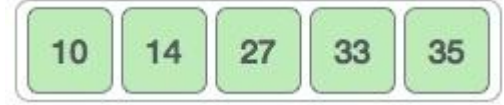
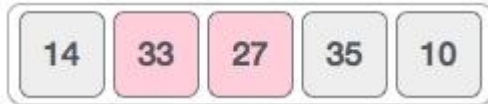
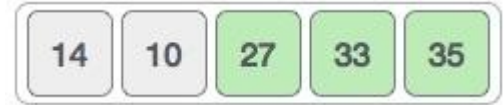
```
int find(int intArray[],int MAX, int data)
{
    int low= 0;
    int high= MAX-1;
    int mid= 0;
    while(low<=high){
        mid=(low+high)/2;
        if(intArray[mid]==data){
            return mid;
        }
        if(intArray[mid]>data){
            high=mid-1;
        }
        else
            low=mid+1;
    }
    return -1;
}
```



Üstel Zaman $O(n^2)$

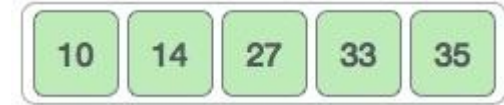


Üstel Zaman $O(n^2)$



Üstel Zaman $O(n^2)$

```
void sort(int intArray[],int MAX)
{
    int temp,i,mov;
    for(mov=0; mov<MAX-1; mov++) {
        for(i=0; i<MAX-1-mov; i++) {
            if(intArray[i]>intArray[i+1]){
                temp=intArray[i];
                intArray[i]=intArray[i+1];
                intArray[i+1]= temp;
            }
        }
    }
}
```



Sorular

