



BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ

DERS 8
Algoritmalar

Algoritma Kavramı

- ▶ Algoritma; bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.

- ▶ Dur komutu alınmadığı sürece aşağıdaki adımları çalıştır:
 - ▶ Bir komut getir
 - ▶ Komutu çöz
 - ▶ Komutu yürüt

Algoritmanın resmi tanımı

- ▶ Algoritma; sonlanan bir işlemi tanımlayan kesin, çalıştırılabilir adımların sıralı bir kümesidir.

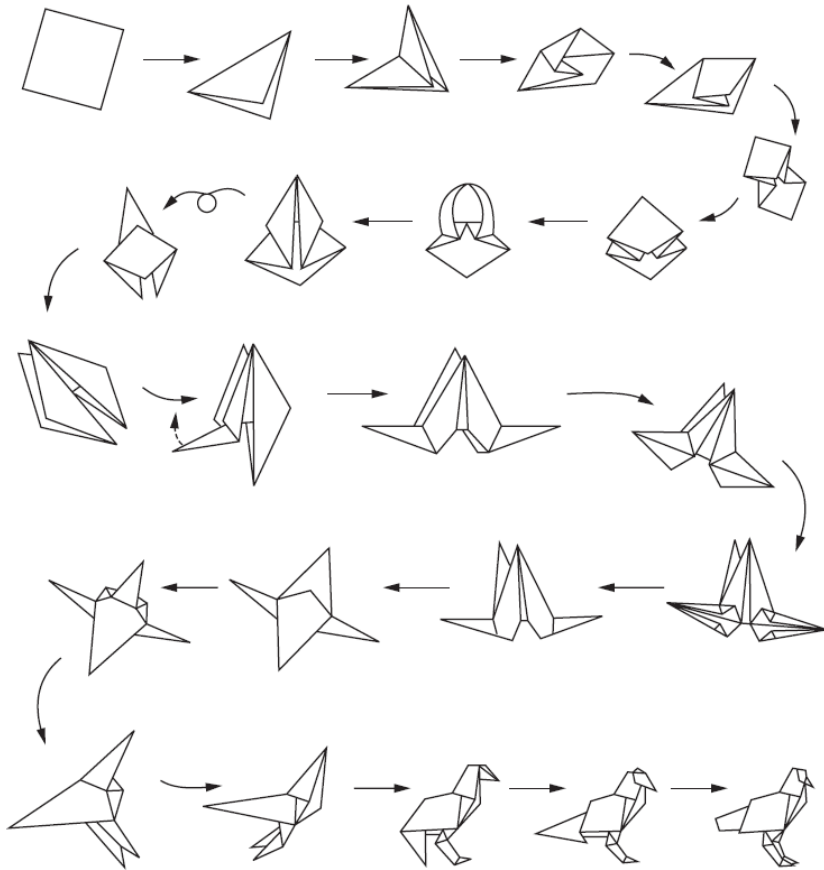
- ▶ Sepette bezelye olduğu sürece devam et:
 - ▶ Sepetten bir bezelye al
 - ▶ Bezelyenin kapsülünü aç
 - ▶ Kapsülden kaseye bezelyeleri dök
 - ▶ Kapsülü at

Algoritmaların soyut doğası

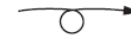
- ▶ $F = (9/5)C + 32$
- ▶ Santigratta okunan dereceyi $9/5$ ile çarp ve daha sonra 32 ekle.

Algoritma gösterimi: Temel Öğeler

- ▶ Bir algoritmanın gösterimi için tanımlanan dil tam olarak tanımlanmadığı zaman iletişim problemleri ortaya çıkar.
- ▶ Temel öğeler (primitifler) bu problemlere iyi tanımlanmış bir dizi yapı taşı oluşturur.



Syntax

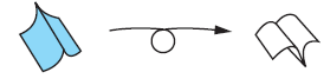


Shade one side of paper



Semantics

Turn paper over
as in

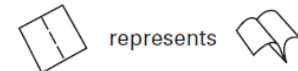


Distinguishes between different sides of paper
as in



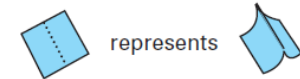
Represents a valley fold

so that



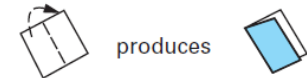
Represents a mountain fold

so that



Fold over

so that



Push in

so that



Sözde kod (Pseudocode)

- ▶ Düşüncelerin algoritma geliştirme sürecinde gayri resmi olarak ifade edilebildiği bir işaret ve rakamlarla gösterme sistemidir.
- ▶ Sözde kod, bilgisayar bilimleri alanında algoritmalar ve programlar oluşturulurken ve aktarılırken kullanılan, günlük konuşma diline benzer ve belli bir programlama dilinin detaylarından uzak anlatımlardır.
- ▶ Genelde kullanılan yol bir programla dilinin söz dizim kurallarını esnetmektir.
- ▶ Hesaplanan değeri saklamak istiyorsak:
isim = ifade
KalanPara = VadesizBakiye + MevduatBakiye

if

- Koşulun doğru veya yanlış olmasına bağlı olarak iki faaliyetten birini seçmek:

if (koşul):
 faaliyet
else:
 faaliyet

```
if (year is leap year):  
    daily total = total / 366  
else:  
    daily total = total / 365
```

```
if (sales have decreased):  
    lower the price by 5%
```

- Eğer GSMH’da artış var ise hisse senedi al aksi halde hisse senedini sat.

while

`while (condition):`
Activity

- ▶ Satmak için biletler olduğu sürece biletleri satmaya devam edin

```
while (tickets remain to be sold):  
    sell a ticket
```

Girintili gösterim

- ▶ = atama; ==karşılaştırma için
- ▶ $X = Y$, $X := Y$, $X == Y$
- ▶ İç içe (nested) if

```
if (not raining):  
    if (temperature == hot):  
        go swimming  
    else:  
        play golf  
else:  
    watch television
```

fonksiyonlar

```
def name():
```

```
    if (. . . ):
        ProcessLoan()
    else:
        RejectApplication()
```

```
def Greetings():
    Count = 3
    while (Count > 0):
        print('Hello')
        Count = Count - 1
```

```
def Sort (List):
```

```
Sort(the organization's membership list)
```

```
Sort(the wedding guest list)
```

- Fonksiyon bir kez oluşturulduktan sonra ihtiyaç duyulduğunda sadece ismi ile çağırılır.

Algoritmanın keşfi

- ▶ Program geliştirme için aşamalar:
 - ▶ Problemi anlamak
 - ▶ Bir algoritmik fonksiyon, problemi nasıl çözebilir fikrini edinmek.
 - ▶ Algoritmayı formüle etmek ve onu bir program olarak göstermek.
 - ▶ Programın doğruluğunu ve diğer problemleri çözmede bir araç olarak potansiyelini değerlendirmek.

Problemi anlama problemi çözmeden gerçekleşmeyebilir.

- ▶ A kişisi B kişinin üç çocuğunun yaşlarını belirlemeden sorumludur.
- ▶ B çocuklarının yaşları çarpımının 36 olduğunu söyler.
- ▶ B çocuklarının yaşları toplamını A'ya söyler.
- ▶ A bir ipucu daha gerek der.
- ▶ B en büyük çocuğunun piyano çaldığını söyler.

Figure 5.5 Analyzing the possibilities

a. Triples whose product is 36

(1,1,36)	(1,6,6)
(1,2,18)	(2,2,9)
(1,3,12)	(2,3,6)
(1,4,9)	(3,3,4)

b. Sums of triples from part (a)

$1 + 1 + 36 = 38$	$1 + 6 + 6 = 13$
$1 + 2 + 18 = 21$	$2 + 2 + 9 = 13$
$1 + 3 + 12 = 16$	$2 + 3 + 6 = 11$
$1 + 4 + 9 = 14$	$3 + 3 + 4 = 10$

İlk adımı atma

- ▶ A, B'nin kazanacağını tahmin etti.
 - ▶ B, D'nin sonuncu olacağını tahmin etti.
 - ▶ C, A'nın üçüncü olacağını tahmin etti.
 - ▶ D, A'nın tahminlerinin doğru olacağını tahmin etti.
-
- ▶ Bu tahminlerden sadece biri doğru ve kazanan tarafından yapıldıysa sıralama nasıl olmuştur.

BASİT DÜŞÜN!

İteratif yapılar

- Bir döngü şeklinde tekrar edilen bir grup talimatı içerir.

Ardışık arama algoritmasının sözde kodu

- Sıralı bir listede arama yapıldığında

Figure 5.6 The sequential search algorithm in pseudocode

```
def Search(List, TargetValue):  
    if (List is empty):  
        Declare search a failure.  
    else:  
        Select the first entry in List to be TestEntry.  
        while (TargetValue > TestEntry and  
            there remain entries to be considered):  
            Select the next entry in List as TestEntry.  
        if (TargetValue == TestEntry):  
            Declare search a success.  
        else:  
            Declare search a failure.
```

Döngü kontrolü

while (condition):

Body

check the condition.
execute the body.
check the condition.
execute the body.

·
·
·

check the condition.

Number = 1

while (Number != 6):

Number = Number + 2

while (the pH level is greater than 4):

add a drop of sulfuric acid

Add a drop of sulfuric acid.

three times, we could write:

Add a drop of sulfuric acid.

Add a drop of sulfuric acid.

Add a drop of sulfuric acid.

`while (condition):`
Activity

`while (there is a coin in your pocket):`
take a coin from your pocket

repeat:
activity
until (condition)

repeat:
take a coin from your pocket
until (there are no coins in your pocket)

Figure 5.8 The while loop structure

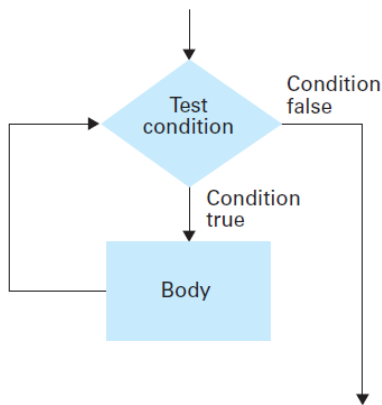
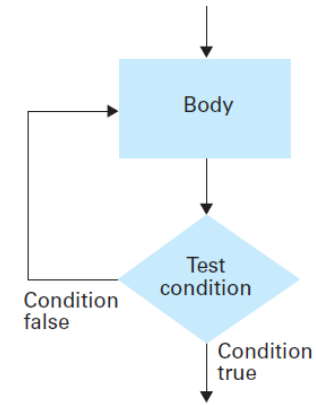


Figure 5.9 The repeat loop structure



For döngüsü

```
for Item in List:
```

```
    . . .
```

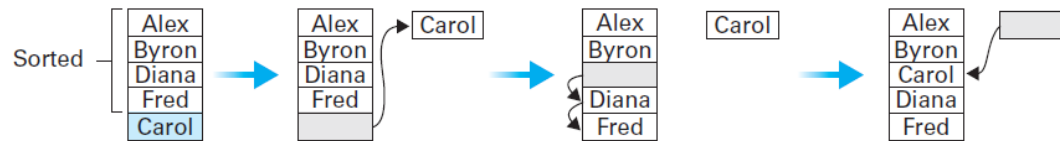
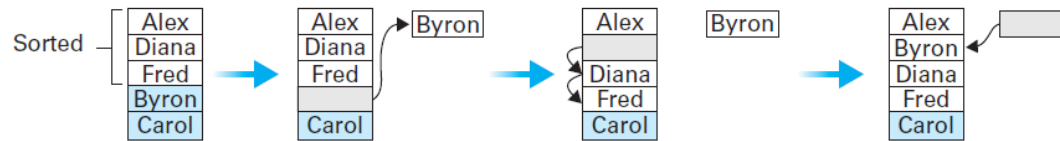
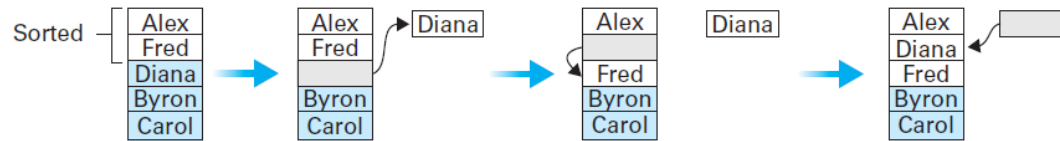
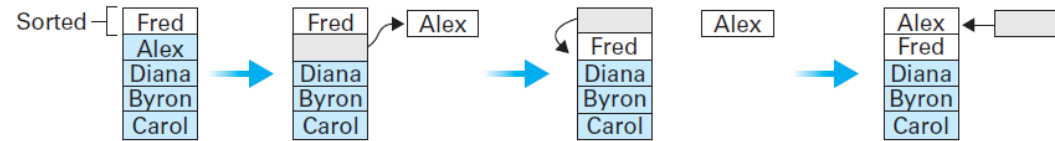
```
Sum = 0
```

```
for Number in List:
```

```
    Sum = Sum + Number
```

Initial list:

Fred
Alex
Diana
Byron
Carol



Sorted list:

Alex
Byron
Carol
Diana
Fred

Eklemeli sıralama algoritması

```
def Sort (List):  
    N = 2  
    while (the value of N does not exceed the length of List):  
        Select the Nth entry in List as the pivot entry.  
        Move the pivot entry to a temporary location leaving  
        a hole in List.  
        while (there is a name above the hole and that name  
            is greater than the pivot):  
            Move the name above the hole down into the hole  
            leaving a hole above the name.  
        Move the pivot entry into the hole in List.  
        N = N + 1
```

Özyinelemeli yapılar

- ▶ Tekrarlanan bir dizi talimatı kendisinin bir alt görevi olarak içerir.
- ▶ Bir döngü yapısında olduğu gibi art arda gerçekleşmez, iç içe gerçekleşir.

İkili arama algoritması

- ▶ Listenin orta elemanı seçilerek aranan değerle kıyaslanır.
- ▶ Büyük veya küçük olmasına göre listenin son kısmı veya ilk kısmı üzerinde aramaya devam edilir.

Original list	First sublist	Second sublist
Alice Bob Carol David Elaine Fred George Harry Irene John Kelly Larry Mary Nancy Oliver	Irene John Kelly Larry Mary Nancy Oliver	Irene John Kelly

İkili arama algoritması

Figure 5.13 A first draft of the binary search technique

```
if (List is empty):  
    Report that the search failed.  
else:  
    TestEntry = the "middle" entry in the List  
    if (TargetValue == TestEntry):  
        Report that the search succeeded.  
    if (TargetValue < TestEntry):  
        Search() the portion of List preceding TestEntry for TargetValue,  
            and report the result of that search.  
    if (TargetValue > TestEntry):  
        Search() the portion of List following TestEntry for TargetValue,  
            and report the result of that search.
```

Figure 5.14 The binary search algorithm in pseudocode

```
def Search(List, TargetValue):  
    if (List is empty):  
        Report that the search failed.  
    else:  
        TestEntry = the "middle" entry in List  
        if (TargetValue == TestEntry):  
            Report that the search succeeded.  
        if (TargetValue < TestEntry):  
            Sublist = portion of List preceding  
                TestEntry  
            Search(Sublist, TargetValue)  
        if (TargetValue > TestEntry):  
            Sublist = portion of List following  
                TestEntry  
            Search(Sublist, TargetValue)
```

Recursive faktoriyel

```
def faktoriyel(n):
```

```
    if n < 1:
```

```
        return 1
```

```
    else:
```

```
        sonuc = n * faktoriyel(n - 1)
```

```
    return sonuc
```

Algoritma verimliliği

Figure 5.18 Applying the insertion sort in a worst-case situation.

Initial list	Comparisons made for each pivot				Sorted list
	1st pivot	2nd pivot	3rd pivot	4th pivot	
Elaine David Carol Barbara Alfred	1 → Elaine David Carol Barbara Alfred	3 → David Elaine 2 → Carol Barbara Alfred	6 → Carol David 5 → Elaine 4 → Barbara Alfred	10 → Barbara Carol 9 → David 8 → Elaine 7 → Alfred	Alfred Barbara Carol David Elaine

Figure 5.19 Graph of the worst-case analysis of the insertion sort algorithm

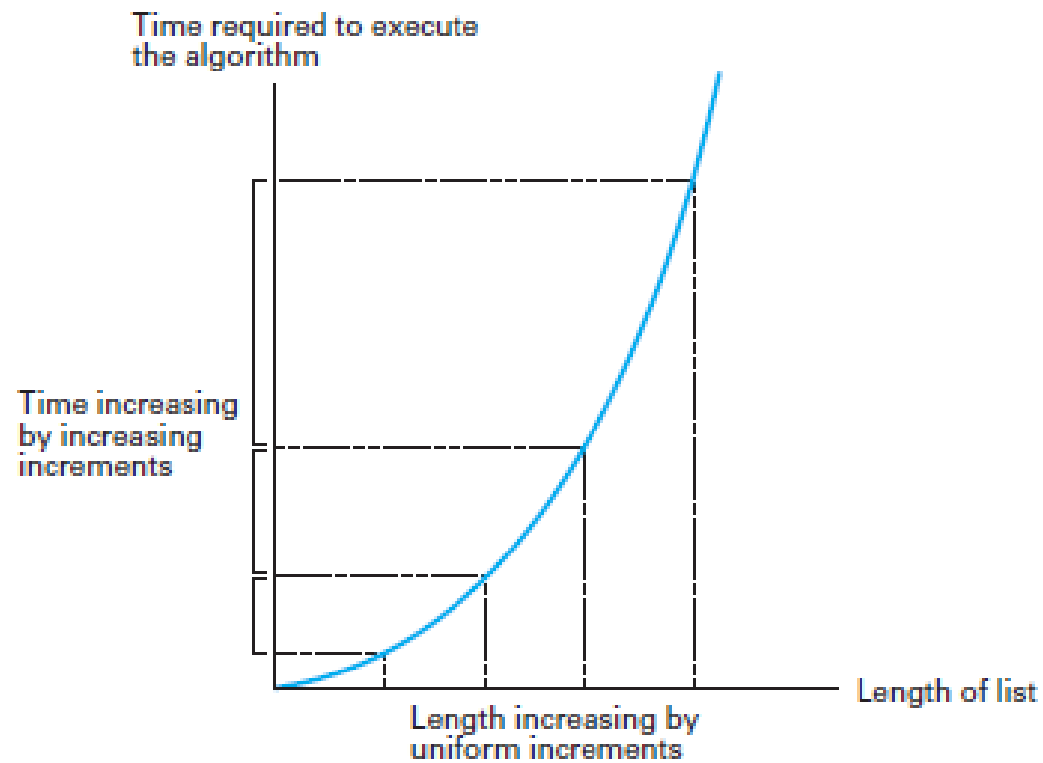
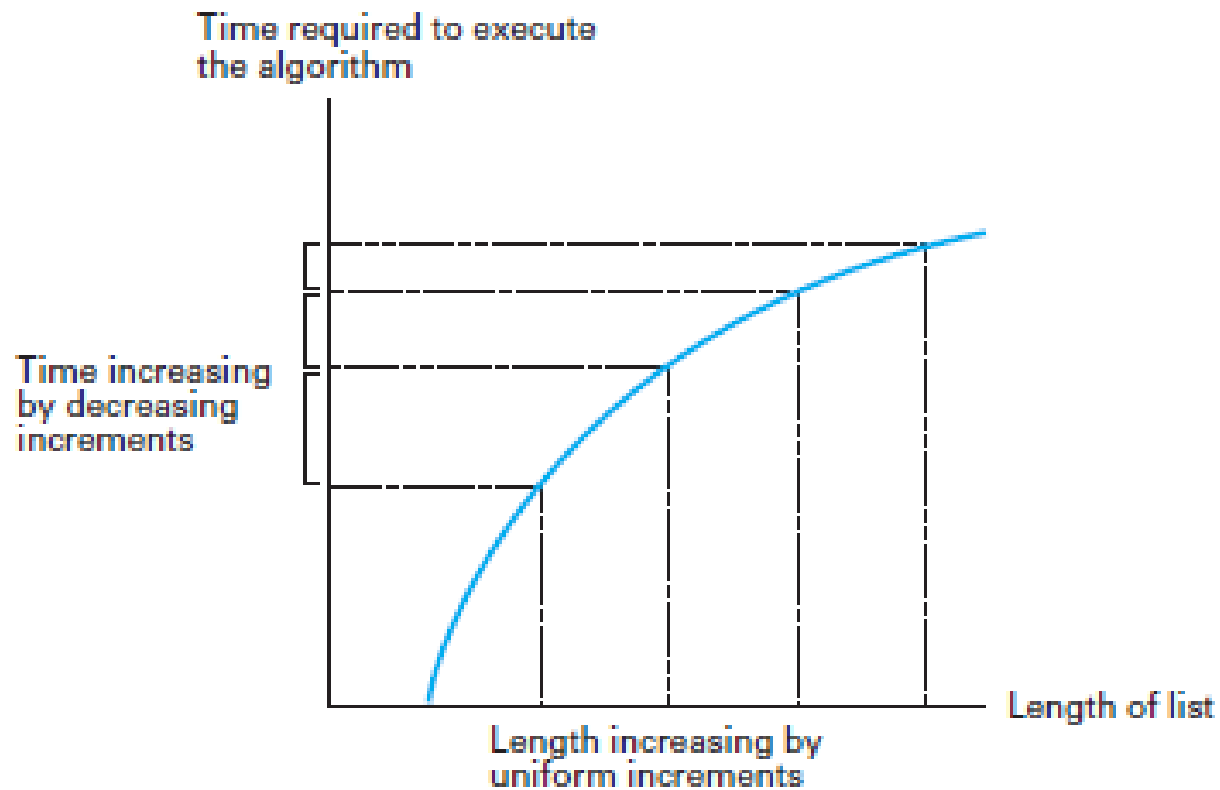


Figure 5.20 Graph of the worst-case analysis of the binary search algorithm



Yazılım doğrulama

Figure 5.21 Separating the chain using only three cuts

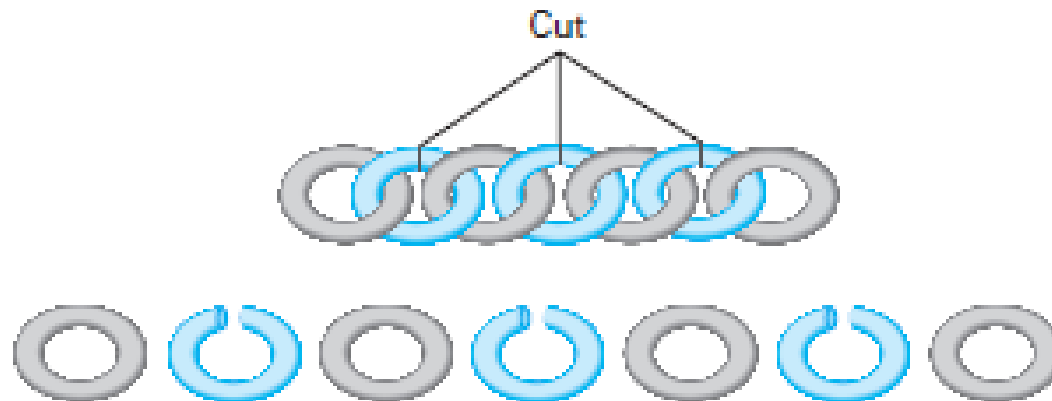
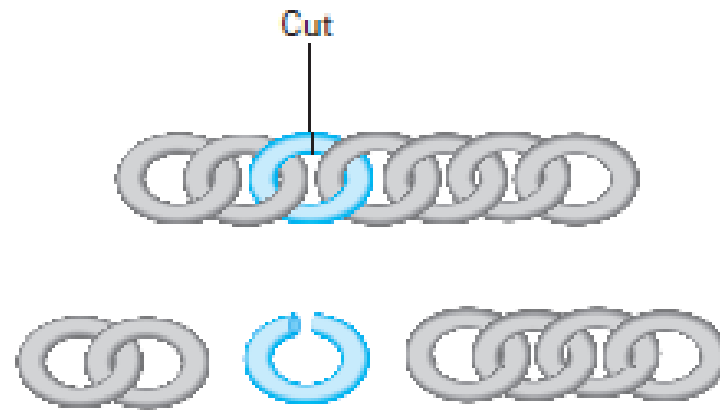


Figure 5.22 Solving the problem with only one cut





Ders bitti

Erciyes Üniversitesi
Selçuk Üniversitesi
Sakarya Üniversitesi
Kahramanmaraş Sütçü İmam Üniversitesi
ders notları kaynak ve içerik olarak kullanılmıştır.