



BTK
AKADEMİ

Programlama

Doç. Dr. Zafer CÖMERT



Bölüm 3

Algoritmalar

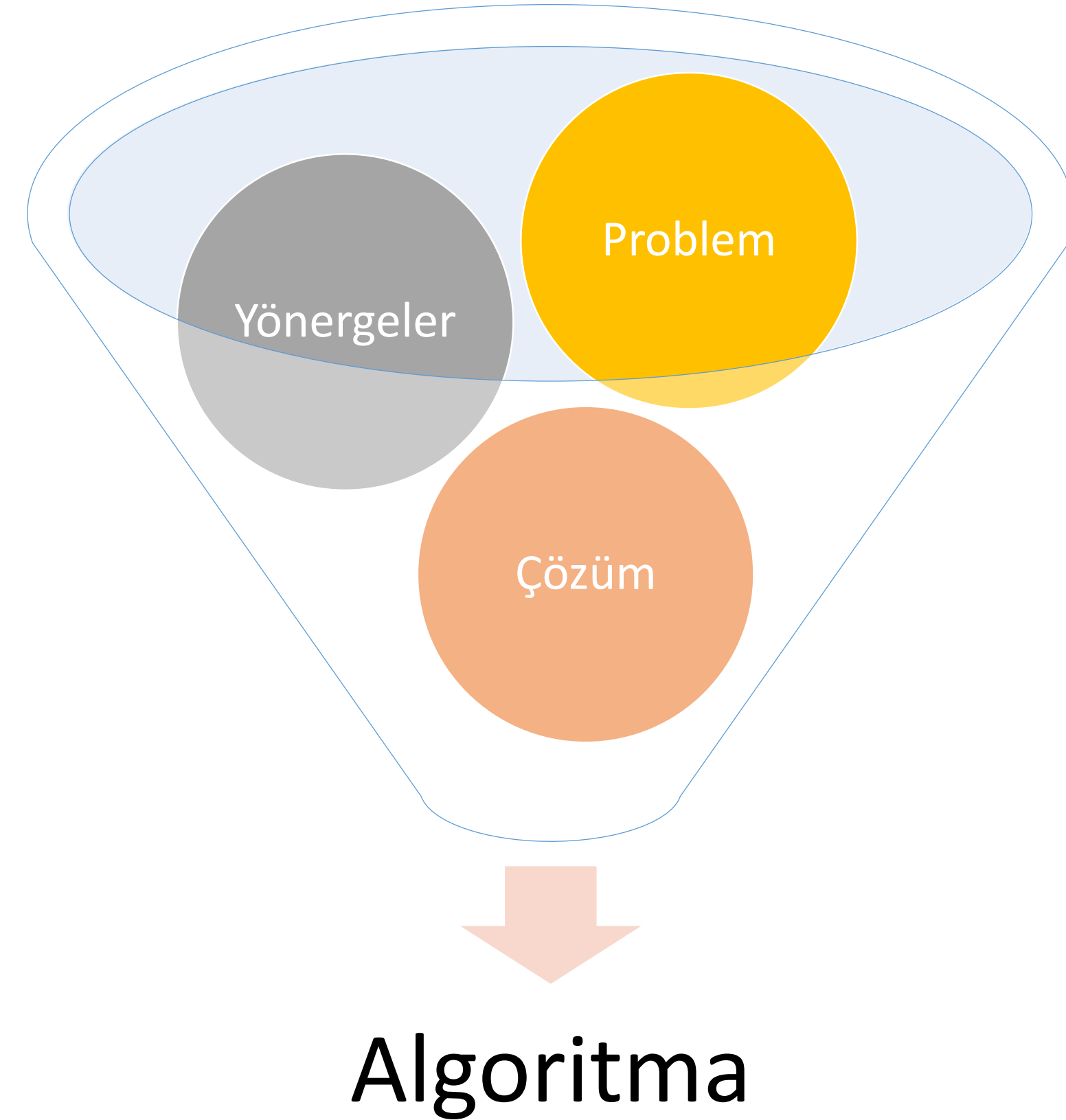
Giriş

İçerik

- Algoritma
- Algoritmaların Özellikleri
- Algoritma Gösterim Yöntemleri
- Algoritma Türleri
- Algoritma Oluşturma Adımları
- Algoritma Tasarımı
- Örnek Algoritma Uygulamaları

Algoritma

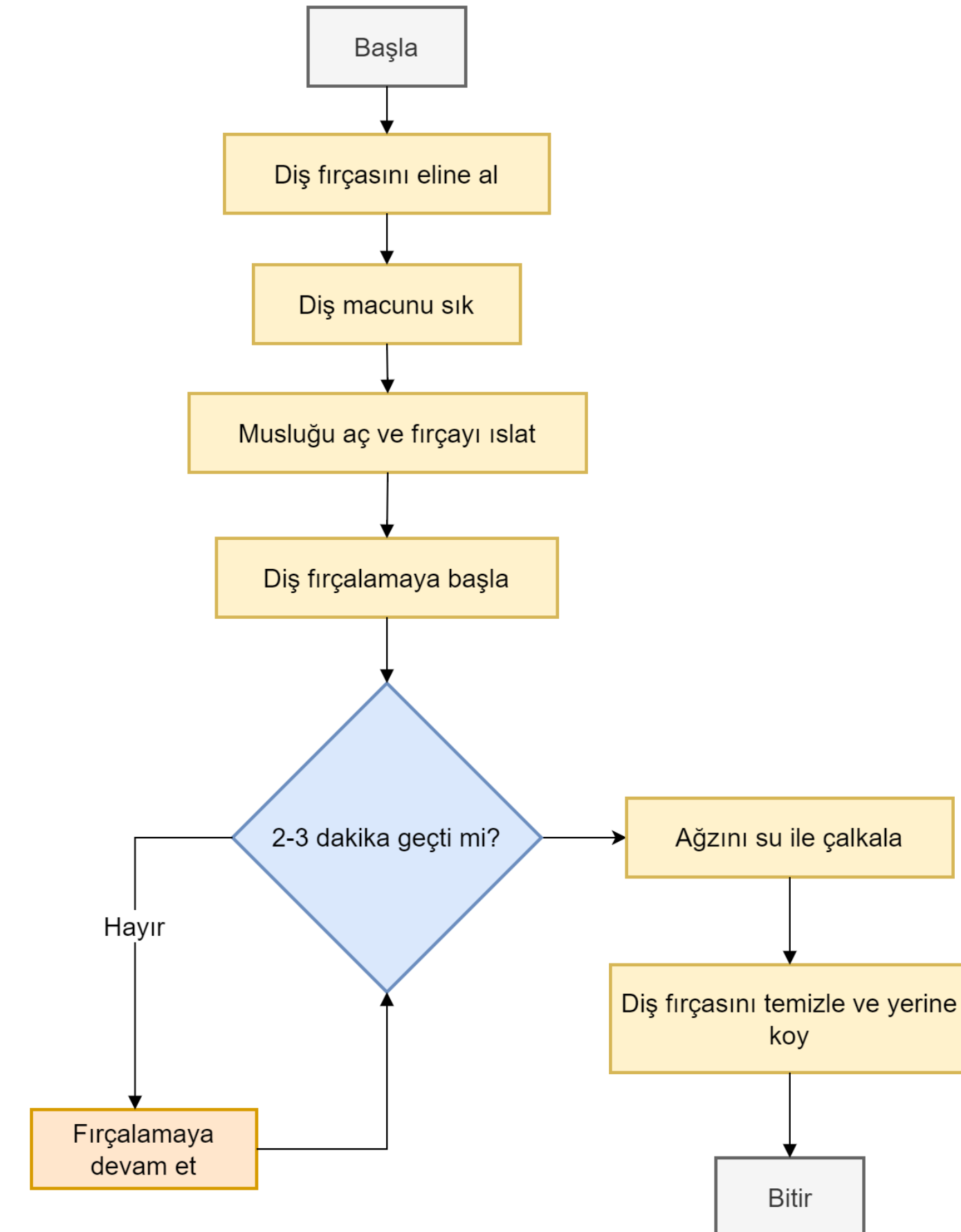
- Algoritma, bir problemi çözmek veya belirli bir amaca ulaşmak için mantıklı ve sıralı adımlardan oluşan bir yol haritasıdır.
- Diğer bir deyişle, algoritma başlangıcı ve bitimi belli olan, adım adım ilerleyen bir süreçtir.



Algoritma

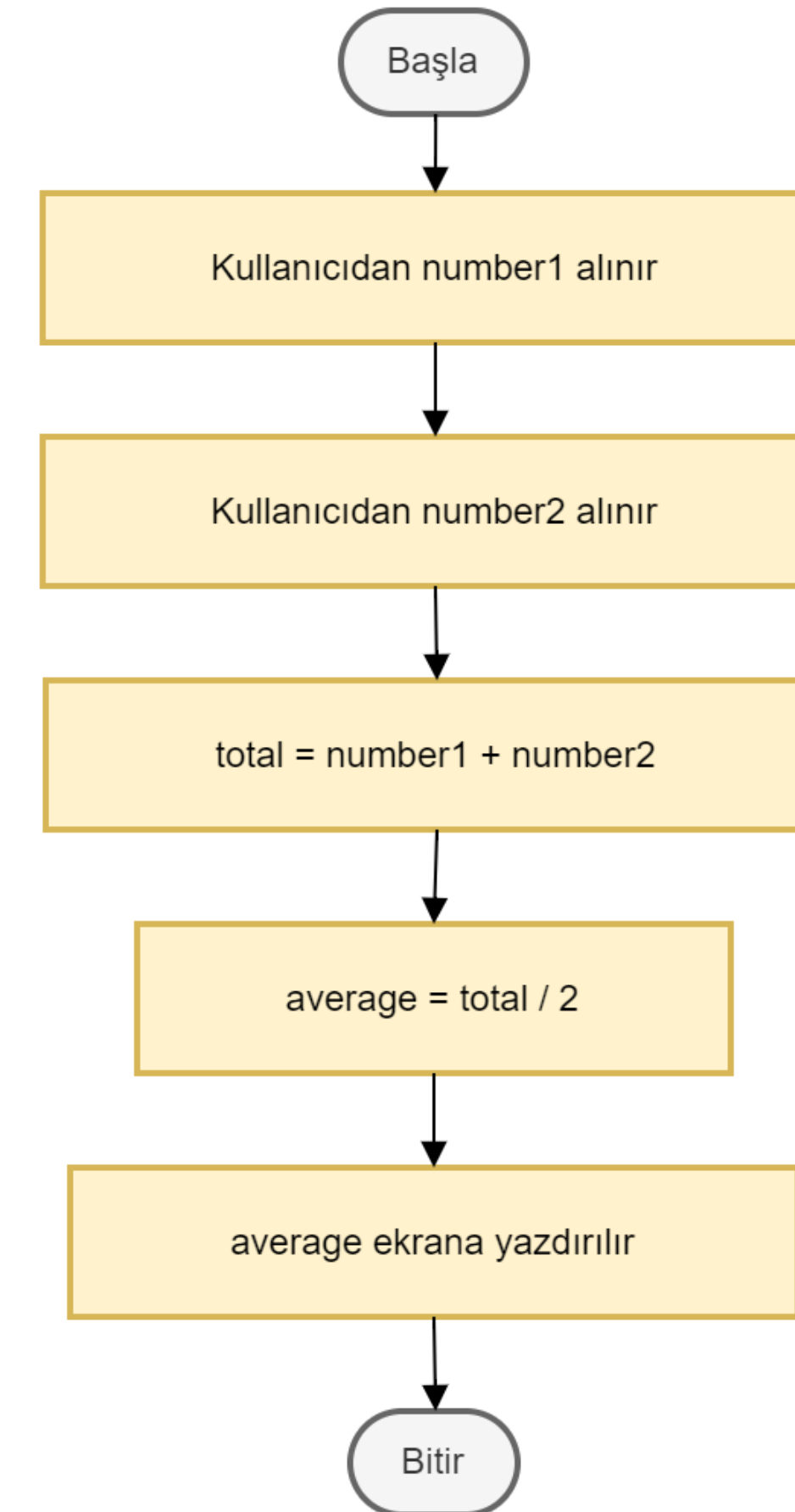
Diş Fırçalama Algoritması

1. Diş fırçasını eline al.
2. Diş fırçasının üzerine diş macunu sık.
3. Musluğu aç ve diş fırçasını hafifçe ıslat.
4. Dişlerini fırçalamaya başla.
5. Dişlerini 2-3 dakika boyunca fırçala.
6. Ağızını su ile çalkala.
7. Diş fırçasını temizle ve yerine koy.
8. Bitir.



Programalama ve Algoritma İlişkisi

- 1) Başla
- 2) Kullanıcıdan iki sayı al,
- 3) Bu sayıları topla,
- 4) Toplamı 2'ye böl,
- 5) Sonucu göster
- 6) Bitir



Programalama ve Algoritma İlişkisi

| Özellik | Algoritma | Programlama |
|-------------|---|---|
| Tanım | Bir problemi çözmek için adım adım belirlenmiş, net ve sıralı işlem dizisi. | Algoritmanın bir programlama dili kullanılarak bilgisayarda çalışacak şekilde kodlanması. |
| Amaç | Problemin çözüm yolunu mantıksal ve sistematik olarak tanımlamak. | Algoritmayı bilgisayarın anlayacağı dile çevirerek çalıştırmak. |
| İfade Şekli | Doğal dil, sözde kod veya akış diyagramı ile ifade edilir. | Programlama dillerinin sözdizimi kullanılarak yazılır (örneğin C++, Java). |
| Uygulama | Soyut ve dil bağımsızdır; her programlama dilinde uygulanabilir. | Belirli bir programlama dili ve ortamında yazılır ve çalıştırılır. |
| İçerik | Problemin çözüm adımları, kararlar, döngüler ve işlemler. | Algoritmanın bilgisayarda işleyebilmesi için gerekli komutlar, değişken tanımları ve dil kuralları. |
| Sonuç | Çözüm yolu ve işlem sırası. | Çalıştırılabilir kod ve program çıktısı. |

Algoritmaların Temel Özellikleri

Adım Adım İfade (Adımlılık)

Başlangıç ve Bitiş

Temel Adımlar

Sonluluk

Kesinlik (Belirlilik)

Algoritma Gösterim Yöntemleri



Doğal Dil (Sözel ifade)



Sözde Kod (Pseudo-code)



Akış Şemaları (Flowchart)

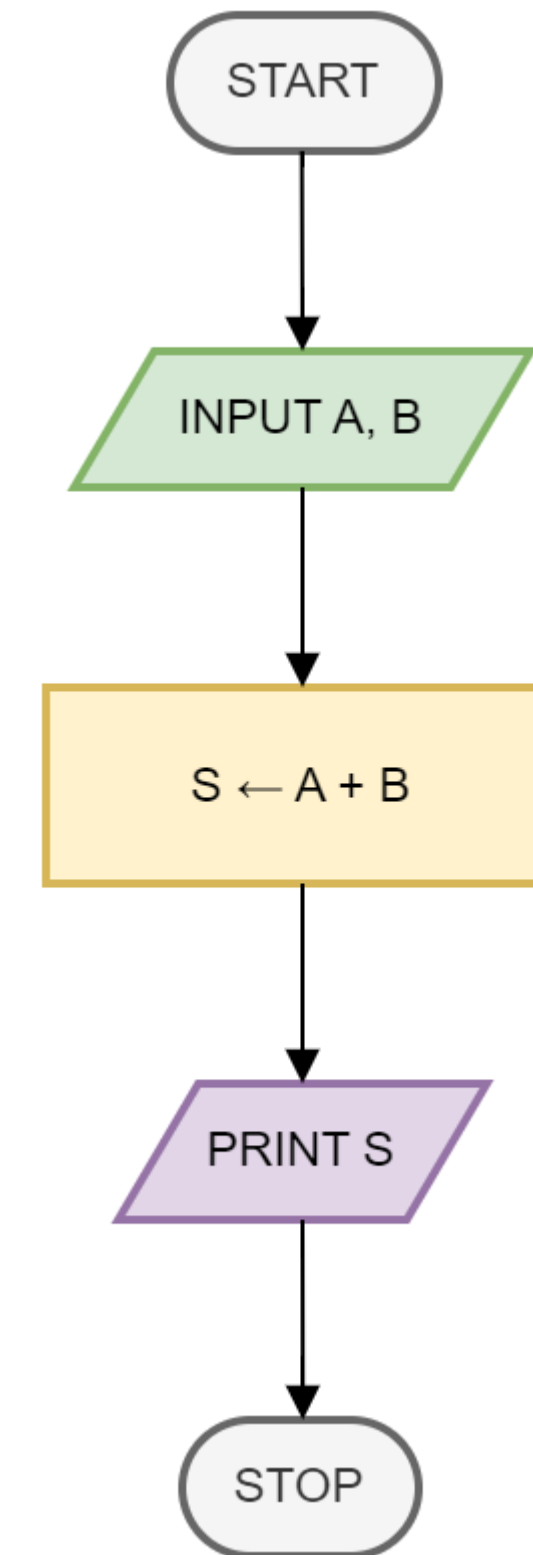
Algoritma Gösterim Yöntemleri

1. Başla,
2. Birinci sayıyı al
3. İkinci sayıyı al
4. Bu iki sayıyı topla
5. Sonucu göster
6. Bitir.

Doğal Dil

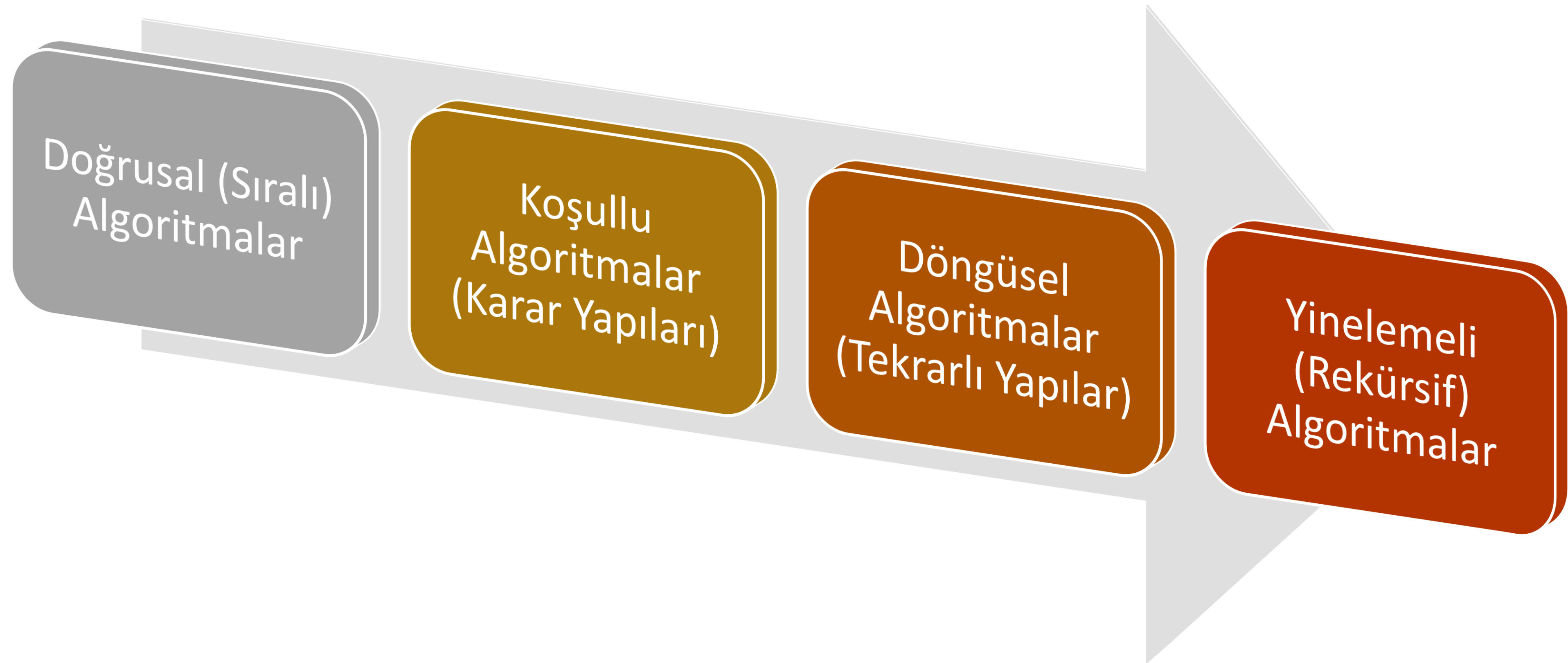
```
BAŞLA  
    say11 = oku()  
    say12 = oku()  
    toplam = say11 + say12  
    yaz(toplam)  
BİTİR
```

Sözde Kod



Akış Şeması

Algoritma Türleri



Doğrusal (Sıralı) Algoritmalar



```
1  # Doğrusal algoritmaya örnek: İki sayının toplamı
2  a = int(input("Birinci sayıyı girin: "))
3  b = int(input("İkinci sayıyı girin: "))
4  toplam = a + b
5  print("Toplam:", toplam)
```

Koşullu Algoritmalar



```
1  # Koşullu algoritma örneği:  
2  sayi = int(input("Bir sayı girin: "))  
3  if sayi >= 0:  
4      print("Pozitif veya sıfır")  
5  else:  
6      print("Negatif")
```

Döngüsel Algoritmalar



```
1  # Döngüsel algoritma
2  for sayi in range(1, 11):
3      print(sayi)
```

Yinelemeli Algoritmalar



```
1 def faktoriyel(n):  
2     # temel durum (sonlandırma koşulu)  
3     if n == 0 or n == 1:  
4         return 1  
5     else:  
6         return n * faktoriyel(n - 1)
```

Algoritma Oluşturma Adımları

Problemi Tanımlama

Girdi ve Çıktıları Belirleme

Çözüm Adımlarını Planlama

Akış Diyagramı Oluşturma (İsteğe bağlı)

Algoritmayı Test Etme

Algoritma Tasarım İpuçları

Problemi İyice Anlayın

Basitten Başlayın

Tüm Olasılıkları Düşünün

Sık Sık Test Edin

Gereksiz Karmaşıklıktan Kaçının

Verimliliği Göz Önünde Bulundurun

Gerekirse Böl ve Fether Paradigmasını Uygulayın

Teşekkürler

ZAFER CÖMERT
Öğretim Üyesi