

ALGORİTMA VE PROGRAMLAMA I

HAFTA 2 ALGORİTMA

Ders İçeriği: Hafta 2

- Problem Nedir?
- Problem Çözme
- Algoritma Nedir?
- Algoritma Gösterim Şekilleri
- Algoritmaların Sınıflandırılması
- Yazılım Geliştirmede Algoritmanın Rolü

Problem Nedir?

➤ Genel Tanım

- Karşılaşılabilecek soruna veya çözülmesi gereken duruma problem denir.
- İnsanlar sürekli düşünür ve problem çözerler.
- Birçok problem basit düşünme ile çözülebilir.

Problem: Bugün ne giysem?

Çözüm: Bunun için muhtemelen pencereden dışarıya bakılır. Hava yağmurlu ise mevsime göre giyinmenin yanı sıra dışarıya çıkarken bir de şemsiye alınması gerekir. Hava güneşli ve sıcak ise daha ince giyinilerek dışarıya çıkılır. Böylece problemin çözümü kendiliğinden oluşturulan bir kararla sağlanır.

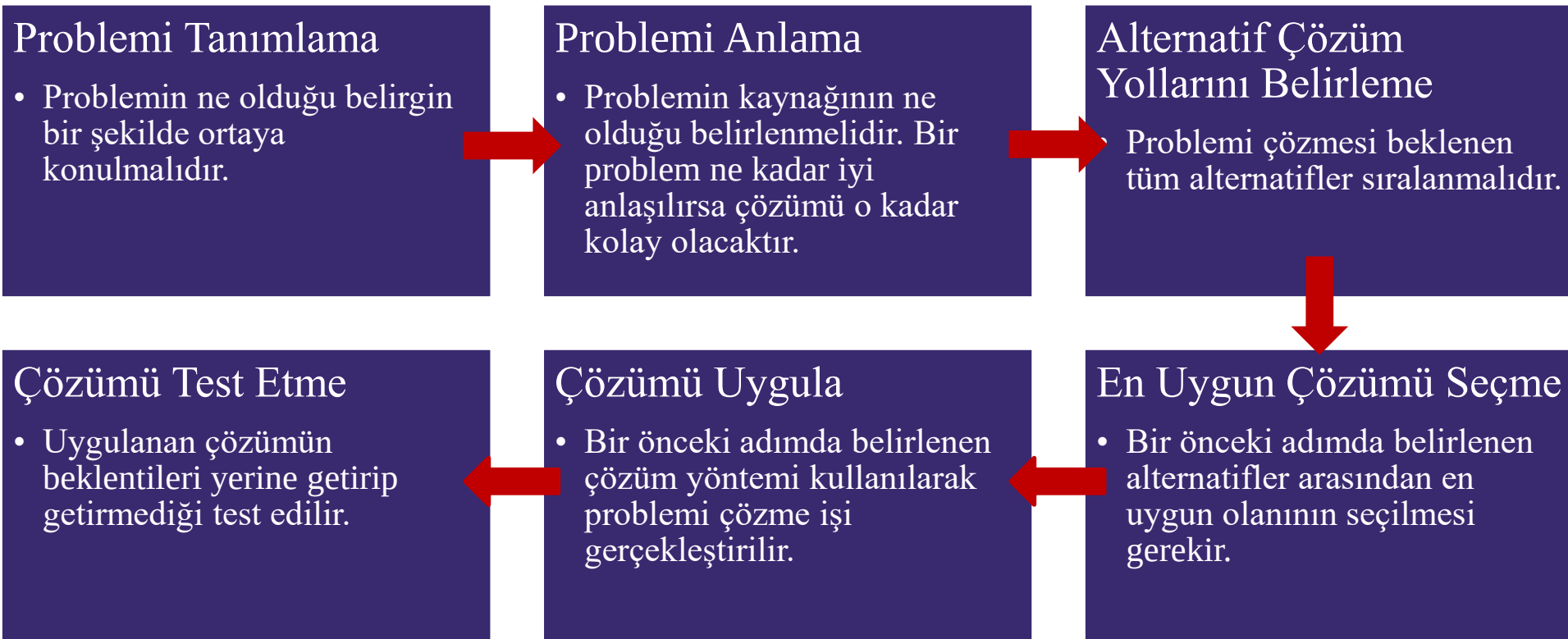
Problem Çözme

➤ Problem Çözme Süreci

- Problemi ortaya koyma aşamasından, problemin çözümünün tamamlanmasına kadar geçen zaman problem çözme süreci olarak adlandırılır.
- Problem çözümünde iki farklı yöntem:
 - Basit problemler deneme-yanılma veya tahminde bulunma ile çözülebilir.
 - Daha karmaşık durumlarda ise problem çözme aşamaları uygulanabilir.

Problem Çözme

➤ Problem Çözme Süreci



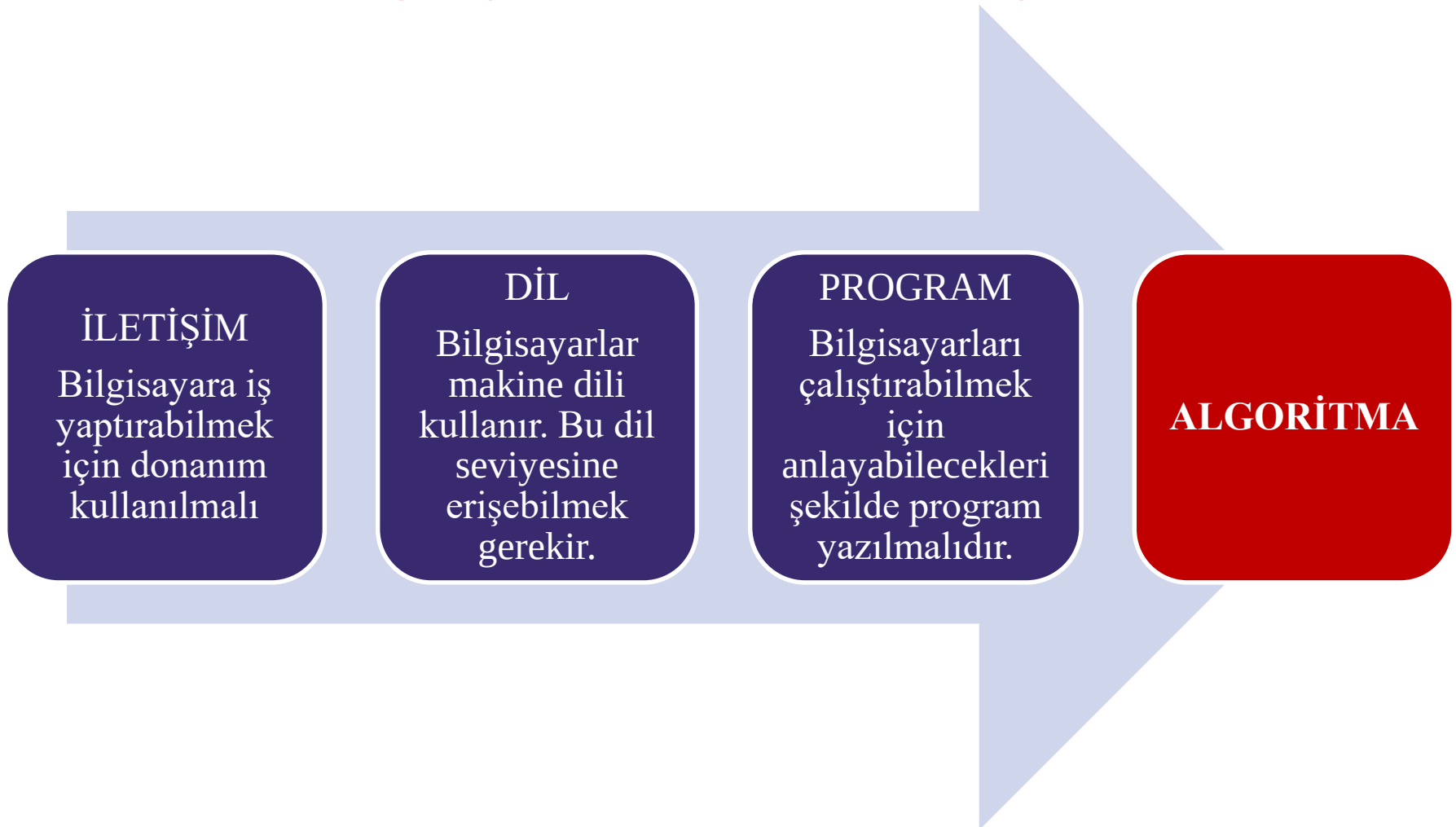
Problem Çözme

➤ Problemi Kim Çözecek?



Problem Çözme

➤ Problemi Bilgisayara Nasıl Çözdüreceğiz?



Algoritma Nedir?

➤ Genel Bakış:

- Algoritma, 800'lü yıllarda yaşamış olan İranlı matematikçi Muhammad ibn Musa al-Khwärizmi'nin yaptığı çalışmalarda ortaya konmuştur.
- İlk algoritma, el-Hârizmî tarafından "Hisab el-cebir ve el-mukabala" kitabında sunulmuştur.
- 12. yüzyılda bu çalışma Latince'ye çevrilirken, çalışmaların sahibi olan el-Hârizmî'nin adından dolayı “algorithm” olarak çevrilmiştir.
- Bu kelime Türkçe'ye ise algoritma olarak girmiştir.
- Tarihçesinden de anlaşılacağı üzere algoritma, matematik alanındaki problemlerin çözümü için kullanılmaktaydı.
- Daha sonra bilgisayarların geliştirilmesiyle bu alandaki problemlerin çözümünde de kullanılmaya başladı.

Algoritma Nedir?

➤ Tanım:

- Algoritma, belli bir problemi çözmek veya belirli bir amaca ulaşmak için tasarlanan yol. Matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.
- Problem çözme yöntemlerinden biri olan algoritma geliştirmek kodlamaya atılan ilk adımdır.
- Algoritma mantığı iyice kavrandıktan sonra bu mantık ile birlikte bir programlama dili kullanılarak yazılım geliştirme süreci başlar.
- Algoritmaların program haline getirilmesi için ise programlama dilleri kullanılır.

Algoritma Nedir?

➤ Algoritmanın Özellikleri:

- https://www.youtube.com/watch?v=cDA3_5982h8&list=PPSV&ab_channel=JoshDarnit

- 1 ▪ Açık ve net olmalıdır.
- 2 ▪ Kullanılacak olan girdiler iyi tanımlanmış olmalıdır.
- 3 ▪ Çıktılar açık ve anlaşılır olmalıdır.
- 4 ▪ Algoritmalar hızlı olmalıdır.
- 5 ▪ Sonlu ve uygulanabilir olmalıdır.

Algoritma Nedir?

- Algoritma oluşturma bir tasarım ürünüdür. Tasarım esnasında göz önüne alınması gereken üç önemli kural vardır.



Algoritma Nedir?

doğru	a ile b sayılarını topla
yanlış	a sayısını büyük bir sayı ile topla

Bir algoritmada yer alan bütün adımlar açık ve kesin bir dille ifade edilmelidir.

doğru	sayıyı bir milyar ile çarp
yanlış	sayıyı bir milyar kez kendisi ile topla

Algoritmada tekrarlardan ve gereksiz bellek kullanımlarından kaçınılmalıdır.

doğru	kullanıcı 3 kez çift sayı girmezse bitir
yanlış	kullanıcı çift sayı girene kadar bekle

Algoritma olası hangi adımlardan ilerlerse ilerlesin bir sonuca varmalıdır.

Algoritma Nedir?

➤ Algoritma Örnekleri

- Yazı yazarken kaleminizin ucu kırıldığında aşağıdaki adımları takip ederek bu sorunu çözebilirsiniz.

1. Kalem tıraşı çıkar.
2. Kalemi al.
3. Çöp kovasının yanına git.
4. Kalemin ucunu aç.
5. Sırana geri dön.
6. Yazmaya devam et.



Algoritma Nedir?

➤ Algoritma Örnekleri

■ Çay demleme algoritması

1. Mutfığa git.
2. Çaydanlığa su koy.
3. Çaydanlığı ocağa koy.
4. Ocağı yak.
5. Su kaynadı mı, kontrol et.
6. Cevap evet ise, çay koy ve demle Cevap hayır ise 6. adıma git.
7. 10 dakika bekle.
8. Çayı doldur.



Algoritma Nedir?

➤ Algoritma Örnekleri

▪ Arabayı çalıştırıp yola çıkma

1. Sürücü koltuğuna geç.
2. Emniyet kemerini tak.
3. Aynaları kontrol et.
4. Anahtarı tak.
5. Contağı çevir.
6. El frenini indir.
7. Vites geç.
8. Gaza bas.



Algoritma Gösterim Şekilleri

➤ Düz Yazı ile Gösterim

- Problemin çözümü adım adım metin olarak yazılır.
- Her satıra bir numara verilir.
- «BAŞLA» ile başlanıp «BİTİR» ile sonlandırılır.
- Örnek: Üçgen alanı hesaplama
 1. BAŞLA
 2. Taban değerini gir
 3. Yükseklik değerini gir
 4. Taban ile yüksekliği çarp ve sonucu ikiye böl
 5. Çıkan sonucu yaz
 6. BİTİR

Algoritma Gösterim Şekilleri

■ Örnek: Tahtaya Adını Yazma Algoritması

1. BAŞLA
2. Yerinden kalk
3. Yönün tahtaya doğru mu?
 - Hayırsa tahtaya dön
 - Evetse 4. adıma git
4. Tahtaya doğru yürü
5. Tahtaya geldin mi?
 - Hayırsa 4.adıma git
 - Evetse 6. adıma git
6. Kalem al
7. Adını yaz
8. BİTİR

Algoritma Gösterim Şekilleri

➤ Sözde Kod (Pseudocode) ile Gösterim

- Herkesin anlayabileceği ve rahatlıkla bir programlama diline çevrilebilecek basit komutlardan oluşan bir dildir.
- Sözde kodun temel işlevi program geliştirmeye geçmeden algoritmayı oluşturmak ve üzerinde tartışabilmektir.
- Sözde kodlar, doğrudan konuşma dilinde ve programlama mantığı altında, eğer, iken gibi koşul kelimeleri ve $> = <$ gibi ifadeler ile beraber yazılır.
- Programda kullanılacak elemanları temsil etmek üzere uygun isimler veya değişkenler seçilir.
- Cebirsel notasyon ve kararlar kullanarak aritmetik işlemler gerçekleştirilir.

Algoritma Gösterim Şekilleri

➤ Sözde Kod (Pseudocode) ile Gösterim

■ Örnek: Üçgen alanı hesaplama

1. BAŞLA
2. t değerini OKU
3. y değerini OKU
4. $A = (t * y) / 2$
5. A değerini YAZ
6. BİTİR

Algoritma Gösterim Şekilleri

■ Örnek: İki Sayının Toplamını Bulma

Düz Yazı	Sözde Kod
1. BAŞLA 2. Birinci sayıyı gir 3. İkinci sayıyı gir 4. İki sayıyı topla 5. Sayıların toplam değerini yaz 6. BİTİR	1. BAŞLA 2. X değerini OKU 3. Y değerini OKU 4. $T = X + Y$ 5. T değerini YAZ 6. BİTİR

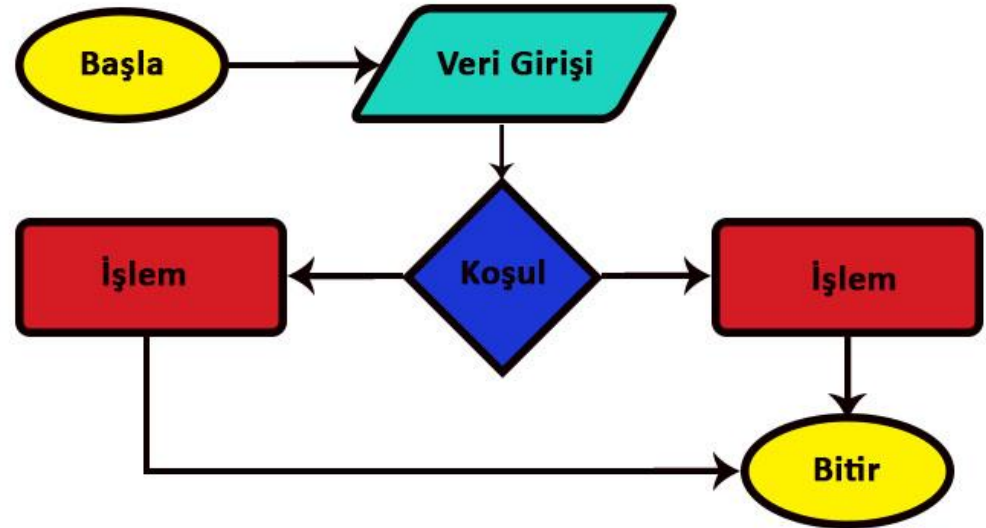
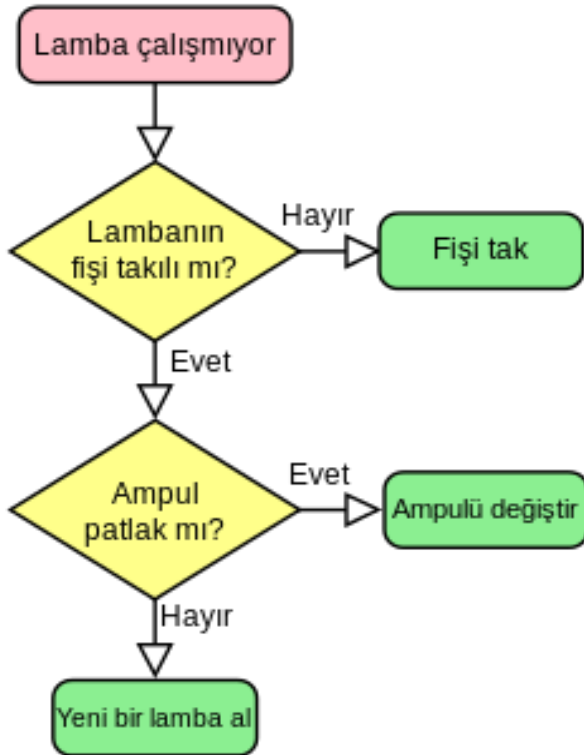
Algoritma Gösterim Şekilleri

➤ Akış Şemaları ile Gösterim

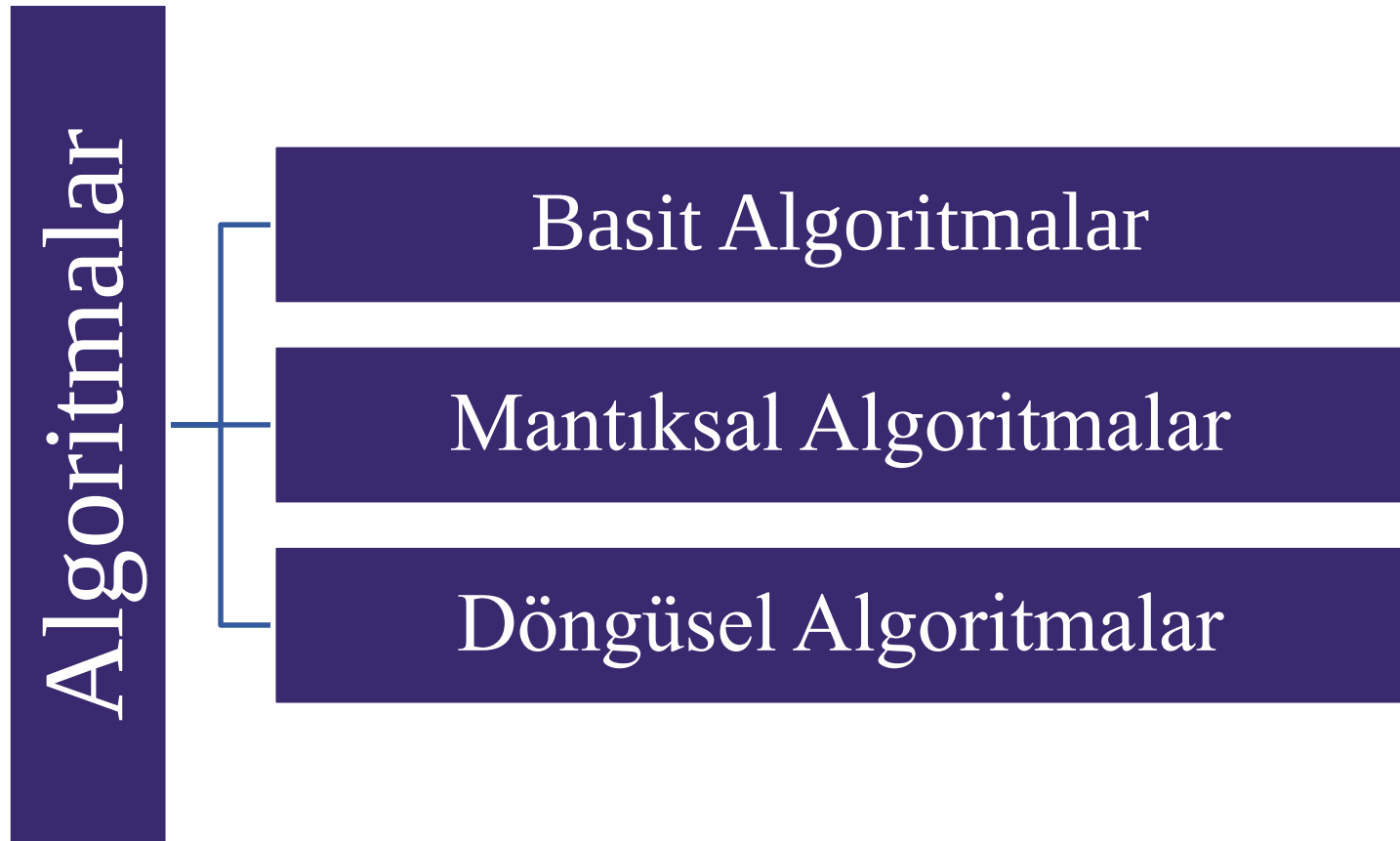
- Bir algoritmanın görsel şekiller ve sembollerle ifade edilmiş haline «Akış Şemaları» veya «Akış Diyagramları» (Flow Chart) adı verilir.
- Akış şeması sembolleri ANSI (American National Standards Institute) standardı olarak belirlenmiştir ve tüm dünyada kullanılmaktadır.
- Algoritma doğal dille yazıldığı için herkes tarafından anlaşılamayabilir ya da başka anlamlar çıkarılabilir.
- Ancak akış şemalarında her bir şekil standart bir anlam taşıdığı için farklı yorumlanması mümkün değildir.

Algoritma Gösterim Şekilleri

➤ Akış Şemaları ile Gösterim



Algoritmaların Sınıflandırılması



Algoritmaların Sınıflandırılması

➤ Basit (Doğrusal, Lineer) Algoritmalar

- İçerisinde mantıksal ifadelerin yer almadığı, program akış dallanmalarının olmadığı algoritmalarlardır.
- Bu algoritmalarda akış düz bir halde baştan sona doğru olacaktır.
- Çoğunlukla küçük hesaplamaları gerçekleştirmek için kullanılırlar.

Algoritmaların Sınıflandırılması

➤ Basit (Doğrusal, Lineer) Algoritmalar

- **Örnek:** Dışarıdan girilen üç adet sayısının toplamını, çarpımını ve ortalamasını hesaplayan algoritma;
 1. Üç adet sayı giriniz; a, b, c
 2. Sayıların toplamını hesaplayınız; $\text{toplam} = a + b + c$
 3. Sayıların çarpımlarını hesaplayınız; $\text{çarpım} = a * b * c$
 4. Sayıların ortalamasını hesaplayınız; $\text{ort} = \text{toplam} / 3$
 5. Sayıların toplamını, çarpımını ve ortalamasını ekrana yazdırınız; $\text{toplam}, \text{çarpım}, \text{ort}$.

Algoritmaların Sınıflandırılması

➤ Mantıksal Algoritmalar

- Algoritma içerisinde mantıksal karşılaştırmaların bulunduğu yapılardır.
- Mantıksal karşılaştırmalara göre algoritmanın akışı farklı adımlara geçecektir.
- Bu şekilde oluşturulan algoritmalara Mantıksal Algoritmalar denir.

Algoritmaların Sınıflandırılması

➤ Mantıksal Algoritmalar

▪ Örnek: Sabah kalkıp kahvaltı hazırlama

1. Yataktan kalk
2. Mutfığa git
3. Eğer ekmek yoksa ekmek al
4. Çayı hazırla
5. Dolaptan kahvaltılıkaları çıkar
6. Bardağın bitince çayını doldur
7. Karnın doyunca sofradan kalk
8. Eğer kahvaltılıkalar bitmişse bulaşık makinesine koy
9. Eğer kahvaltılıkalar bitmemişse kahvaltılıkaları dolaba koy
10. Sofrayı temizle

Algoritmaların Sınıflandırılması

➤ Mantıksal Algoritmalar

- **Örnek:** Girilen üç adet sayı içinden en büyük sayıyı bulan algoritma
 1. Üç adet sayı giriniz; a,b,c
 2. En büyük sayı a olsun; $eb=a$
 3. Eğer b en büyükten büyük ($b>eb$) ise en büyük b olsun; $eb=b$
 4. Eğer c en büyükten büyük ($c>eb$) ise en büyük c olsun; $eb=c$
 5. En büyük sayıyı ekrana yazdır; eb

Algoritmaların Sınıflandırılması

➤ Döngüsel Algoritmalar

- Program için geliştirilen algoritmada bir işlem birden fazla tekrar ediyorsa döngülü algoritma yapısı kullanılır.
- Döngüsel algoritmalarda mantıksal karşılaştırma yapısı özel olarak kullanılır.
- Eğer algoritma içerisinde kullanılan mantıksal karşılaştırma işlemi sonucunda programın akışı karşılaştırma yapılan yerden daha ileriki bir adıma değil de daha önceki adıma gidiyorsa bu şekilde oluşturulmuş algoritmalara döngüsel algoritma denir.
- Yani döngüsel algoritmalarda mantıksal karşılaştırma sonucunda program daha önceki adımlara gider.

Algoritmaların Sınıflandırılması

➤ Döngüsel Algoritmalar

▪ Örnek: Pasta yapımı

1. Gerekli malzemeleri hazırla
2. Yağı ve şekeri kaba koy ve çırp
3. Yumurtayı kaba kır ve çırp
4. Karışıma un koy ve çırp
5. Karışımı kek kalıbına koy
6. Pastayı fırına koy
7. 10 dk. bekle
8. Pişmiş mi kontrol et
 - a. Pişmiş ise 9. adıma git
 - b. Değilse 7. adıma dön
9. Pastayı fırından çıkar ve fırını kapat.

Algoritmaların Sınıflandırılması

➤ Döngüsel Algoritmalar

▪ Örnek: Faktöriyel hesaplama algoritması

1. Başla
2. Bir n sayısı girin
3. Faktoriyel = 1
4. Sayi = 0
5. Eger Sayi $> n - 1$ ise 9.Adıma git.
6. Sayi = Sayi + 1
7. Faktoriyel=Faktoriyel*Sayi
8. Adım 5'e tekrar git
9. Faktoriyeli yazdır
10. Bitir

Algoritmaların Sınıflandırılması

➤ Döngüsel Algoritmalar

- **Örnek:** İki sayının en büyük ortak bölenini bulma algoritması
 1. Başla
 2. Sayıları giriniz; a, b
 3. Eğer $a > b$ ise b'yi a'dan çıkar ve tekrar a'ya yaz ($a = a - b$) ve Adım 3'e git
 4. Eğer $b > a$ ise a'yı b'den çıkar ve tekrar b'ye yaz ($b = b - a$) ve Adım 3'ye git
 5. Eğer $a = b$ ise adım 6'e git
 6. En büyük bölen a değerini ekrana yaz; a
 7. Bitir

Yazılım Geliştirmede Algoritmanın Rolü

Analiz

- Problemin Tanımlanması

Tasarım

Algoritma

- Problemin Çözümüne Yönelik İşlem Adımları

Programlama

- Algoritmanın Programlama dili ile komutlar dizisi olarak tanımlanması

Test

Bakım

Yazılım Geliştirmede Algoritmanın Rolü

➤ Algoritma Oluşturmada Dikkat Edilmesi Gerekenler

- Algoritma tasarlanırken bütün problem parçaları aynı anda çözülmeye çalışılmaz.
- Problem kendi içinde alt parçalara (alt problemlere) bölünür.
- Alt parçalar ayrı ayrı çözümlenerek asıl problemin çözümü gerçekleştirilir.
- Bu yöntem yukarıdan aşağıya tasarım (topdown design) ya da böl ve yönet (divide & conquer) yaklaşımı olarak adlandırılır.

Yazılım Geliştirmede Algoritmanın Rolü

➤ Örnek: Bana bir Eczane programı lazım?

- Problemi anlamak için küçük parçalara ayırmamız lazım.
 - Sağlık sektöründe eczanecilik nasıl işliyor?
 - İlaç giriş-çıkış süreçleri nasıl gerçekleşiyor?
 - Satacak ürünü bitince depodan nasıl mal alırım?
 - Müşterilere satış nasıl yapılıyor?
 - Reçetesiz ürünlerin satışında fark var mı?
 - Sağlık Bakanlığı e-recete entegrasyonu nasıl?
 - Barkod okuma nasıl gerçekleşiyor?
- Tüm bu alt problemlerin çözümü ayrı ayrı ele alınarak birleştirilmelidir.