

Algoritma ve Akış Şeması

1. Algoritma Nedir?

Algoritma, bir problemin çözümünü sağlayan sıralı ve mantıksal adımlar bütünüdür. Bilgisayar bilimi, mühendislik, tıp, finans ve daha birçok alanda karmaşık süreçleri çözmek için algoritmalar kullanılır. Temel amacı, bir sonuca ulaşana kadar yapılması gereken adımları tanımlamaktır. Bu adımlar kesin, uygulanabilir ve sonlu sayıda olmalıdır.

İyi bir algoritmanın taşıması gereken özellikler:

- Giriş (input) ve çıkış (output) açıkça tanımlanmalıdır.
- Her adımı anlaşılır ve net olmalıdır.
- Her durumda çalışmalı ve sonlu adımda sona ermelidir.
- Etkili ve verimli olmalıdır.

2. Algoritmanın Amacı Nedir?

Algoritmalar, çeşitli işlemleri sistemli bir şekilde gerçekleştirmek için kullanılır.

Amaçlar:

- Sorunları küçük parçalara ayırarak çözmek
- Süreçleri sistematikleştirmek
- Bilgisayar programlarının mantıksal temelini oluşturmak
- Yazılım geliştirmenin planlama aşamasını gerçekleştirmek
- Hataları önceden tespit edip maliyeti azaltmak
- İnsan hatasını minimize ederek otomasyon sağlamak

3. Yazılı Algoritma Örnekleri (Kodlama Dili Olmadan)

3.1. Çay Koyma Algoritması

- Çaydanlığa yeterli miktarda su koy.

2. Ocağın altını aç.
3. Su kaynayınca alt demliğe dök.
4. Üst demliğe kuru çay ekle.
5. Kaynar suyu üst demliğe dök.
6. 5 dakika demle.
7. Fincana dök ve servis et.

3.2. Araba Çalıştırma Algoritması

1. Aracın içine otur.
2. Anahtarı kontağa tak.
3. Emniyet kemerini tak.
4. Vitesi boşa al.
5. Fren pedalına bas.
6. Anahtarı çevirerek marşa bas.
7. Motor çalışırsa vitesi bire al ve yavaşça hareket et.

3.3. Uçak Bileti Alma Algoritması

1. Gidilecek tarih ve yer belirlenir.
2. Uçak bileti uygulaması açılır.
3. Gidiş-dönüş bilgileri girilir.
4. Uygun seferler listelenir.
5. Fiyat karşılaştırması yapılır.
6. Seçilen uçuşa kişisel bilgiler girilir.
7. Ödeme yapılır, bilet mail olarak alınır.

3.4. Toplama/Çıkarma Uygulaması

1. Kullanıcıdan işlem türü alınır (topla/çıkar).
2. Birinci sayı alınır.
3. İkinci sayı alınır.
4. İşleme göre işlem yapılır.
5. Sonuç kullanıcıya gösterilir.

3.5. Finansal İşlem Algoritması (Aylık Bütçe)

1. Aylık gelir alınır.
2. Sabit giderler (kira, fatura vb.) girilir.
3. Değişken giderler (market, ulaşım vb.) girilir.
4. Toplam gider hesaplanır.
5. Gelir - gider = tasarruf hesaplanır.
6. Tasarruf pozitifse öneri, negatifse uyarı verilir.

4. Akış Şemaları (Metinsel Biçimde)

4.1. Çay Koyma

[Başla] → [Su koy] → [Su kaynat] → [Demliğe çay koy] → [Su ekle] → [5 dk bekle] → [Çayı dök] → [Bitir]

4.2. Araba Çalıştırma

[Başla] → [Araca otur] → [Anahtarı tak] → [Kemer tak → Vites boşa → Fren]

→ [Anahtarı çevir → Motor çalıştı mı?]

→ Hayır → [Bitir]

→ Evet → [Vites al → Sürüş] → [Bitir]

4.3. Uçak Bileti Alma

[Başla] → [Tarih ve yer seç] → [Uygulama aç] → [Bilet ara → Seç → Bilgi gir] → [Ödeme yap → Mail al] → [Bitir]

4.4. Toplama/Çıkarma Uygulaması

[Başla] → [İşlem türü seç] → [Sayı 1 al] → [Sayı 2 al]

→ Topla mı?

→ Evet → [Topla → Göster] → [Bitir]

→ Hayır → [Çıkar → Göster] → [Bitir]

4.5. Finansal İşlem (Bütçe)

[Başla] → [Gelir ve giderleri al → Toplamla]

→ [Gelir - Gider → Pozitif mi?]

→ Hayır → [Uyarı ver] → [Bitir]

→ Evet → [Tebrik et + Göster] → [Bitir]

5. Python Kod Örnekleri

5.1. Toplama/Çıkarma Uygulaması

```
islem = input("İşlem türü (topla/çıkara): ")
sayi1 = int(input("1. sayıyı girin: "))
sayi2 = int(input("2. sayıyı girin: "))

if islem == "topla":
    print("Sonuç:", sayi1 + sayi2)
elif islem == "çıkara":
    print("Sonuç:", sayi1 - sayi2)
else:
    print("Geçersiz işlem!")
```

5.2. Finansal Bütçe Hesaplama

```
gelir = float(input("Aylık gelir: "))
gider = float(input("Aylık toplam gider: "))
kalan = gelir - gider

if kalan >= 0:
    print(f"Tebrikler! Tasarrufunuz: {kalan} TL")
else:
    print("Uyarı: Giderler, gelirinizi aşıyor!")
```

6. Akış Şeması Nedir?

Akış şeması, bir algoritmanın veya sürecin adımlarını sembollerle ifade eden grafiksel bir temsildir. Problemleri daha kolay anlamayı ve çözmeyi sağlar. Her sembol belirli bir işlevi temsil eder.

7. Akış Şeması Sembolleri

Sembol	Anlamı
Oval	Başla/Bitiş

Dikdörtgen	İşlem yapılacak adım
Paralelkenar	Girdi/Çıktı işlemleri
Elmas	Karar yapısı (if/else)
Ok	Akış yönünü belirtir

8. Algoritma Analizi Nedir?

Algoritma analizi, bir algoritmanın verimliliğini ölçmek için yapılan değerlendirmedir.

- **Zaman Karmaşıklığı:** Algoritmanın çalışması için geçen süre. Örn: $O(n)$, $O(n^2)$
- **Alan Karmaşıklığı:** Bellekte kapladığı yer.

Örnek:

- Bubble Sort $\rightarrow O(n^2)$
- Merge Sort $\rightarrow O(n \log n)$

Veri büyüdükçe daha verimli algoritmalar seçilmelidir.

9. Algoritma Okuma Nedir?

Yazılmış bir algoritmayı adım adım inceleyip ne yaptığını anlamaktır.

Adımlar:

- Girdi ve çıktıyı belirle
- Adım adım işlemleri analiz et
- Koşul ve döngü yapılarını yorumla
- Sonucu tahmin et

Örnek Kod:

```
x = 10
y = 20
if x < y:
    z = x + y
else:
    z = x - y
print(z)
```

Çıktı: 30

10. Sonuç

Algoritmalar ve akış şemaları, yazılım geliřtirmenin temel yapı taşlarıdır. İşlemler mantıksal olarak algoritmalarla modellenir, akış şemalarıyla görselleştirilir ve Python gibi dillerle uygulanabilir hale gelir. Bu bilgi birikimi, yazılım geliřtiricilerin analitik düşünme ve problem çözme becerilerini artırır.

Bu metin, BTK Akademi "Bilgi Teknolojilerine Giriş" eğitiminden edinilen bilgilerle **Arda Karadağ** tarafından hazırlanmıştır.