



Bilgisayar Mühendisliğine Giriş

Yrd.Doç.Dr.Hacer KARACAN

ALGORİTMA VE AKIŞ ŞEMALARI

- Algoritmada olması gereken özellikler
- Algoritma tasarımı
- Akış şemaları

Algoritma Tanımı

- Algoritma, herhangi bir sorunun çözümü için izlenecek yol anlamına gelmektedir.
- Çözüm için yapılması gereken işlemler hiçbir alternatif yoruma izin vermeksizin sözel olarak ifade edilir.
 - Diğer bir deyişle algoritma, verilerin, bilgisayara hangi çevre biriminden girileceğinin, problemin nasıl çözüleceğinin, hangi basamaklardan geçirilerek sonuç alınacağını, sonucun nasıl ve nereye yazılacağını sözel olarak ifade edilmesi biçiminde tanımlanabilir.
- Algoritma hazırlanırken, çözüm için yapılması gerekli işlemler, öncelik sıraları göz önünde bulundurularak ayrıntılı bir biçimde tanımlanmalıdırlar.

Algoritmada olması gereken özellikler

- Etkin ve genel olma
- Sonlu olma
- Yanılmazlık
- Giriş/Çıkış tanımlı olma
- Başarım

Soru Çözme Adımları

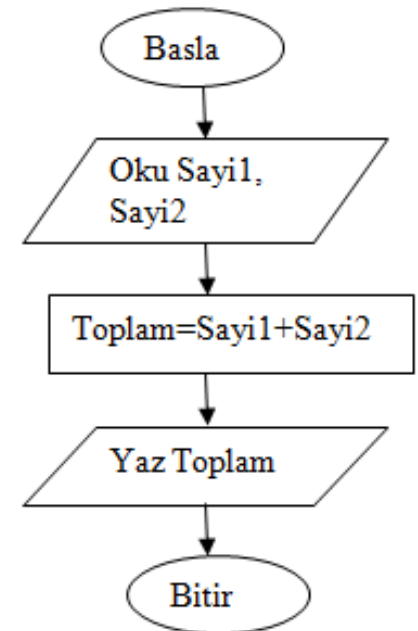
- Bilgisayar ortamında bir problem çözülürken aşağıdaki adımlara dikkat edilmelidir:
 - Soru Tanımlama
 - Algoritma Geliştirme
 - Girdi ve Çıktı Biçimi Belirleme
 - Akış Şemasını Çizme
 - Kodlama
 - Programı Sınama
- Programlamanın en önemli kısmı **problemin çözümü için algoritma geliştirebilmektir.**
 - Algoritma hazırlandıktan sonra hazırlanan algoritmanın herhangi bir programlama dilinde kodlanması işin en basit kısmıdır.

Örnek - I

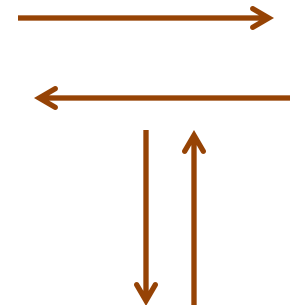
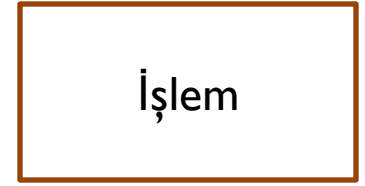
- Klavyeden girilen 2 sayının toplamının bulunması:
- Yapılacak olan işlemler :
 - Klavyeden 2 adet sayı girilecek. O zaman bellekte 2 odacık açmalıyız. Yani veri girişi için 2 tane değişken kullanmak zorundayız.
 - Klavyeden girilen ve ram belleğe aktarılan bu iki değişken toplanacak. Toplam sonucunu yine ram belleğe aktarılmalıdır. Bu yüzden ram bellekte bir odacık daha açmalıyız.
 - Bulduğumuz toplam sonucunu ekrana yazdırmalıyız. Aksi taktirde toplam sonucu sadece ram bellekte bulunur ve sonucu kullanıcı göremez.

- Algoritma:

- Adım 1: Basla
- Adım 2: Oku Sayı 1, Sayı2
- Adım 3: $\text{Toplam} = \text{Sayı 1} + \text{Sayı2}$
- Adım 4: Yaz Toplam
- Adım 5: Bitir

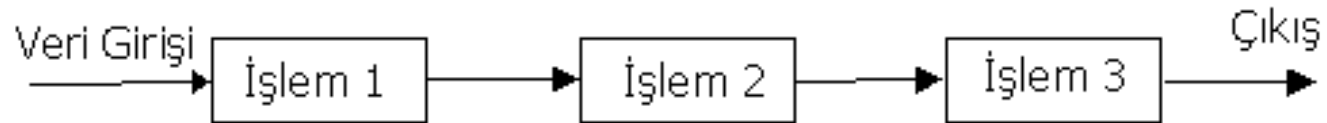


Akış Şeması Simgeleri

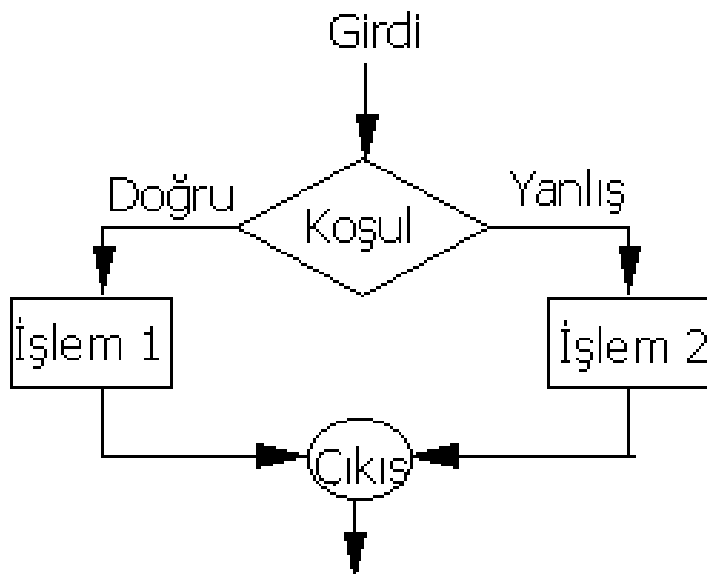


Akış Şemaları

- **Sıralı Yapı**

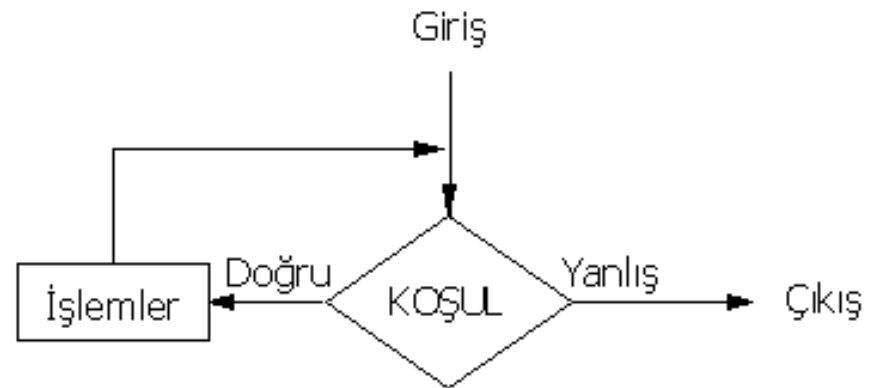
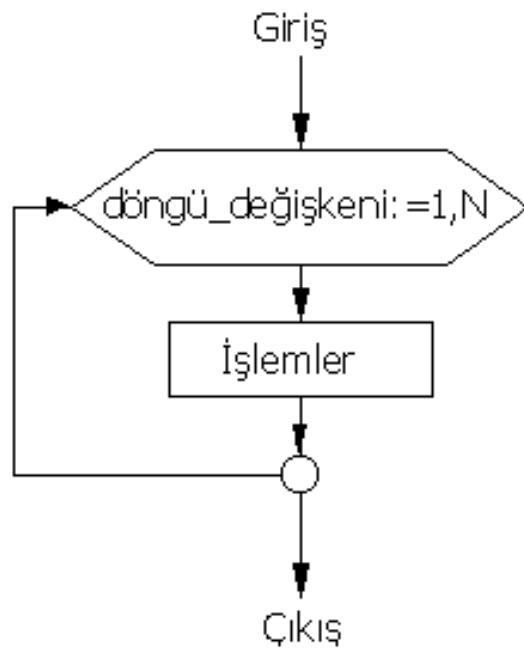


- **Karar Verme Yapısı**



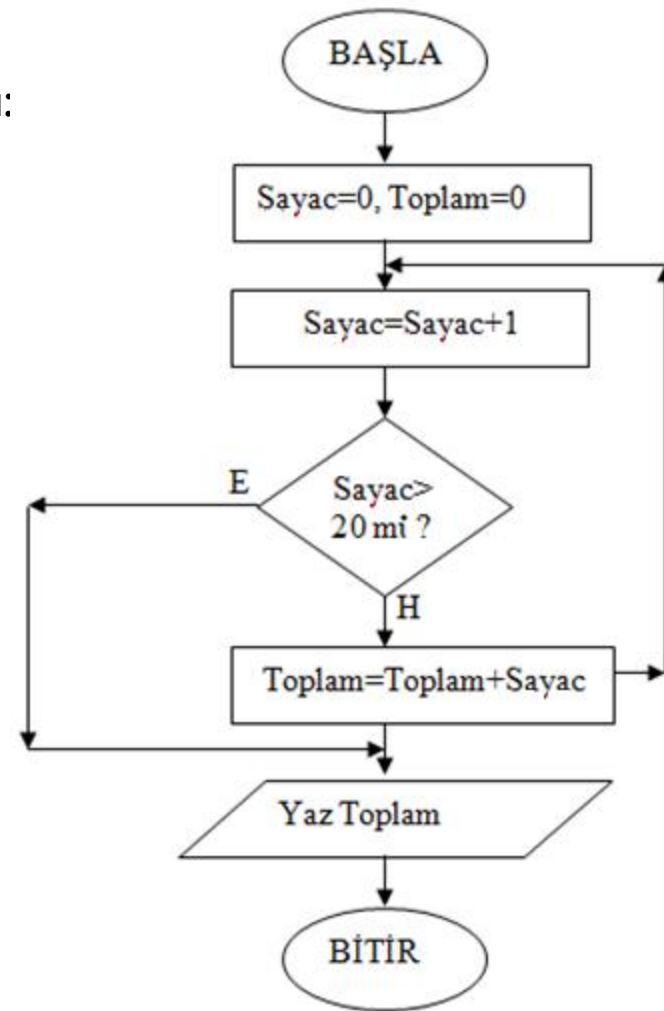
Akış Şemaları

- Tekrarlı Yapı**



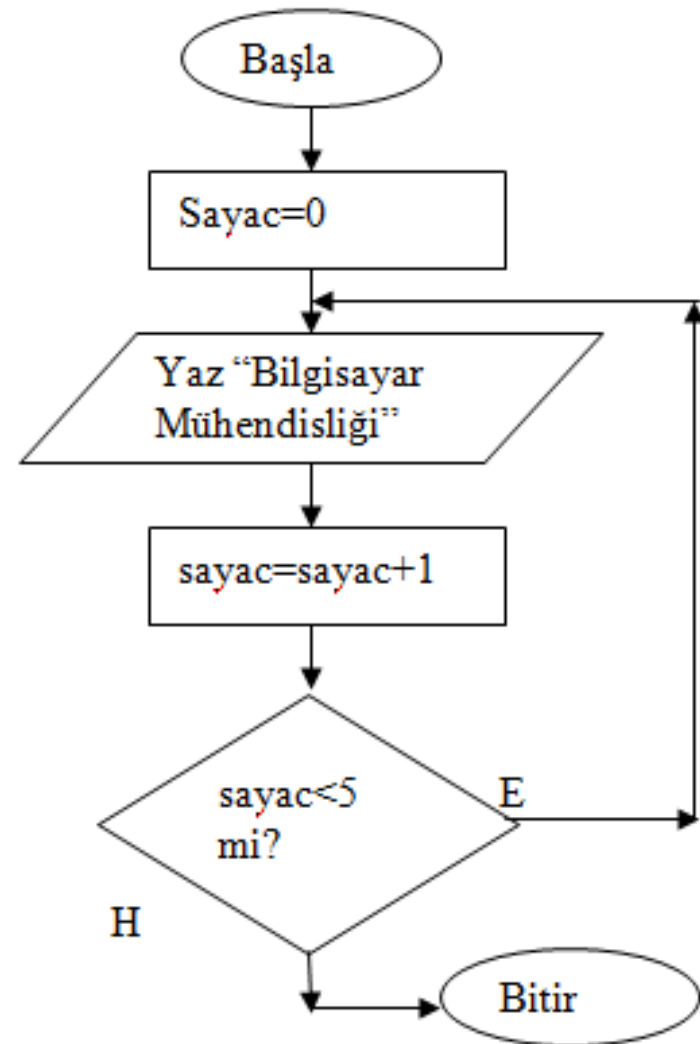
Örnek - 2

- 1-20 arasındaki tamsayıların toplamının bulunması:
 - Adım1.BAŞLA
 - Adım2. Sayac=0,Toplam=0
 - Adım3. Sayac=Sayac+1
 - Adım4. EĞER Sayac>20 İSE GİT 7
 - Adım5. Toplam=Toplam+Sayac
 - Adım6. GİT 3
 - Adım7.YAZ “1-20 Arası Sayıların Toplamı=”,Toplam
 - Adım8. BİTİR



Örnek - 3

- Bir cümlenin 5 kez ekrana yazdırılması:
 - Adım1.BAŞLA
 - Adım2. Sayac=0
 - Adım3.YAZ “Bilgisayar Mühendisliği”
 - Adım4. Sayac=Sayac+1
 - Adım5. EĞER Sayac<5 İSE GİT 3
 - Adım6. BİTİR



Kaba (Pseudo) Kod

- Kaba-kod → Algoritmanın yarı programlama dili kuralları, yarı konuşma diline dönük olarak tanımlanması
- Gerçek kod → algoritmanın herhangi bir programlama diliyle gerçekleştirilmiş hali

Kaba (Pseudo) Kod - Örnek

Algorithm BINARY SEARCH

```
proc binary__search

    /* The  $n$  records of the file are ordered in in-
       creasing order of the keys. */

1    LOWER := 1
2    UPPER :=  $n$ 
3    while LOWER  $\leq$  UPPER do
4        MIDDLE :=  $\lfloor (LOWER + UPPER) / 2 \rfloor$ 
5        if key[sought] = key[MIDDLE]
6            then terminate successfully.
7        else if key[sought] > key[MIDDLE]
8            then LOWER := MIDDLE + 1
9        else UPPER := MIDDLE - 1
10       end
11       terminate unsuccessfully.
end binary__search
```

Algoritma ve İlişki Tanımlaması için Çeşitli Yöntemler

- **Metinsel Tanımlama**
 - Algoritma adımlarının yazılı ifadesi
- **Akış Şeması**
 - Algoritma adımlarının şematik ifadesi
- **N-S (Nassi-Schneiderman) Şemaları**
 - Tüm işlemler/akış bir dikdörtgen içerisinde gösterilir
- **W-O (Warnier-Orr) Diyagramları**
 - Kaba-kod benzeri şekilsel olmayan bir yöntem
- **Bachman Notasyonu**
 - İlişkisel veritabanı veya bilgi sistemleri için veri modeli diyagramları oluşturmak amacıyla kullanılır
- **“Crow’s Foot Spec” Dili**
 - Veri modelleme ve veritabanı tasarımında kullanılır
- **“IDEFIXERD Spec” Dili**
 - Veri yapısı tasarımı, mantıksal veritabanı tasarımı, ilişki diyagramları çiziminde kullanılır