

35-ALGORİTMİK YAPI

Algoritmada Mantık Bağlaçları

Dikkat: Algoritmalar, bilgisayar biliminde kullanılan adım adım çözüm yöntemleridir. Bu süreçte mantık işlemlerinde doğru (1) ve yanlış (0) değerleri kullanılır.”

- VE BAĞLACI (\wedge)
- VEYA BAĞLACI (\vee)
- YA DA BAĞLACI (\vee)
- İSE BAĞLACI (\Rightarrow)
- ANCAK VE ANCAK BAĞLACI (\Leftrightarrow)

1. VE BAĞLACI (\wedge)

Tüm koşulların doğru olduğu durumlarda doğru, diğer durumlarda yanlış olan bağlaçtır.

İki durumda doğru ise sonuç doğru, diğer durumlarda sonuç yanlıştır.

Günlük hayattan bir örnek:

Bir Öğrencinin Not ortalaması 49 dan büyük ve Devamsızlık sayısı 20 den küçük olması gerekiyor.

Burada

Doğru Sonuca: 1 diyoruz

Yanlış Sonuca: 0 diyoruz

Ortalama	İşlem	Devamsızlık		Sonuç
0	\wedge	0	\equiv	0
0	\wedge	1	\equiv	0
1	\wedge	0	\equiv	0
1	\wedge	1	\equiv	1

2. VEYA BAĞLACI (\vee)

Birden fazla şartta birinin doğru olduğu durumları kontrol eder.

İki durum yanlış ise sonuç yanlış, diğer durumlarda sonuç doğrudur.

Örnek:

Bir Öğrencinin yazılı notu 49 dan büyük veya proje teslim etmesi gerekiyor.

Yazılı Notu	İşlem	Proje		Sonuç
0	\vee	0	\equiv	0
0	\vee	1	\equiv	1
1	\vee	0	\equiv	1
1	\vee	1	\equiv	1

3. YA DA BAĞLACI ($\underline{\vee}$)

Ya da bağlacı iki ifadeden yalnızca birinin doğruluğunu kontrol eder.

İki durumdan birisi doğru ise doğru diğer durumlarda yanlıştır.

Örnek: Sadece 1 tanesini seç

- *Yazılı geçti, performans yok* → ✓ geçer
- *Yazılı kötü, performans var* → ✓ geçer
- *İkisi de var* → ✗ kabul edilmez
- *İkisi de yok* → ✗ kalır

Yazılı Notu	İşlem	Proje		Sonuç
0	$\underline{\vee}$	0	\equiv	0
0	$\underline{\vee}$	1	\equiv	1
1	$\underline{\vee}$	0	\equiv	1
1	$\underline{\vee}$	1	\equiv	0

4. İSE BAĞLACI (\Rightarrow)

Bir şart gerçekleştiğinde bu şarta bağlı olarak bir sonucun ortaya çıkacağını belirtmek için kullanılır.

Birinci doğru ikinci yanlış ise yanlış, diğer durumlarda doğrudur.

Örnek: Yağmur şemsiye ilişkisi

Yağmur	İşlem	Şemsiye		Sonuç
0	\Rightarrow	0	\equiv	1
0	\Rightarrow	1	\equiv	1
1	\Rightarrow	0	\equiv	0
1	\Rightarrow	1	\equiv	1

Sadece yağmurun yağıp şemsiye almadığımız durum yanlıştır.

5. ANCAK VE ANCAK BAĞLACI (\Leftrightarrow)

İki koşulun birbirine bağlı olduğunu ifade etmek için kullanılır.

İki durum da aynı ise sonuç doğru, farklı ise sonuç yanlıştır.

- Biri doğruysa diğeri de doğru olmalı ✓
- Biri yanlışsa diğeri de yanlış olmalı ✓

Örnek : Bir öğrenci takdir belgesi alır ancak ve ancak not ortalaması 85 ve üzeri ise.

Örnek : Bir öğrenci takdir belgesi alamaz ancak ve ancak not ortalaması 85 ve altı ise.

Ortalama	İşlem	Belge		Sonuç
0	\Leftrightarrow	0	\equiv	1
0	\Leftrightarrow	1	\equiv	0
1	\Leftrightarrow	0	\equiv	0
1	\Leftrightarrow	1	\equiv	1

Ö: “Bir üçgenin tüm kenar uzunlukları eşittir ancak ve ancak tüm açı ölçüleri 60 dereceyse”

Ö: $A = \{1, 3, 5, 7, 9\}$

$B = \{2, 3, 5, 11, 13\}$

Kümelerine göre aşağıda verilen bileşik önermeleri sağlayan değerleri bulunuz.

***I.* $(x \in A) \wedge (x \in B)$**

***II.* $(x \in A) \vee (x \in B)$**

Ö:

İsim	Çeşit	Boy	Fiyat (TL)
CEMİL	Sade	Büyük	250
AYŞE	Çilek	Orta	200
İREM	Limon	Büyük	225
EMİNE	Çilek	Küçük	150
EMRE	Sade	Orta	200
ERCAN	Çikolata	Orta	220

Tablo: Bir arkadaş grubunun dondurma tercihleri

Tabloda bir kafeye giden 5 kişilik bir arkadaş grubunun dondurma siparişlerinin bilgisi verilmiştir.

Buna göre

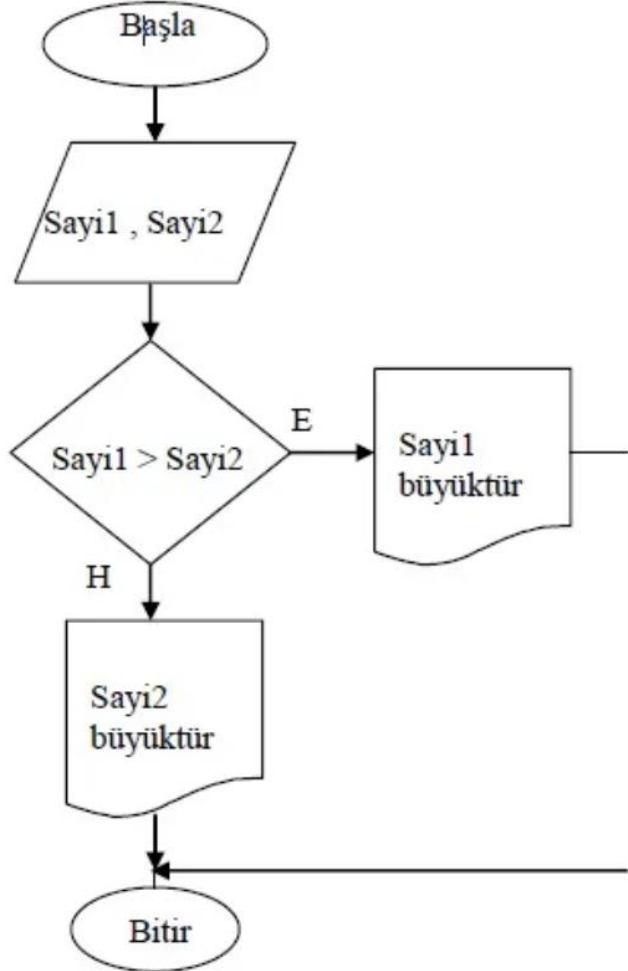
I. $(Orta\ boy) \vee (Sade)$

II. $(Dondurma\ sade) \vee (Fiyat > 200)$

önergelerini sağlayan kişileri bulunuz.

Algoritmada Döngü

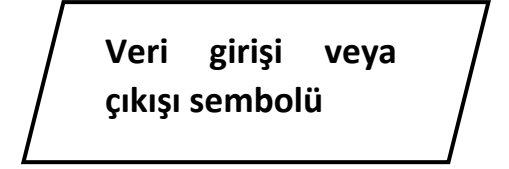
Akış şeması bir problemin çözümünü belirli bir sırayla, semboller ve oklar kullanarak gösteren işlem adımlarıdır.



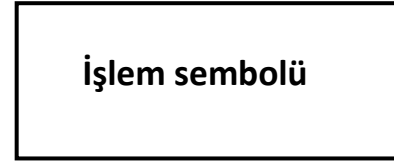
Kullanılacak şekil ve semboller aşağıda verilmiştir.



Başla-Bitir
sembolü



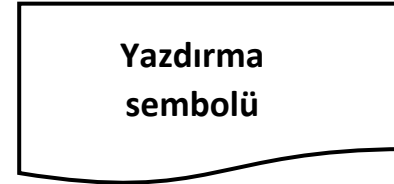
Veri girişi veya
çıkışı sembolü



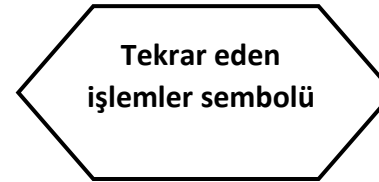
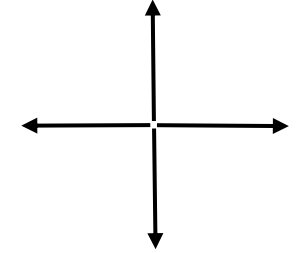
İşlem sembolü



Karar
sembolü



Yazdırma
sembolü



Tekrar eden
işlemler sembolü



Birleştirici ya da
bağlantı
noktaları

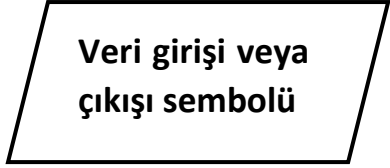
ALGORİTMADA AKIŞ ŞEMASI



Başla-Bitir
sembolü



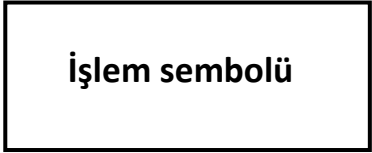
Akış şemasının
başlama ve bitişini
gösterir.



Veri girişi veya
çıkışı sembolü



Girilen verileri veya
algoritma sonunda
çıkan verileri
gösterir.



İşlem sembolü



İşlem yapmak için
kullanılan
semboldür



Karar
sembolü



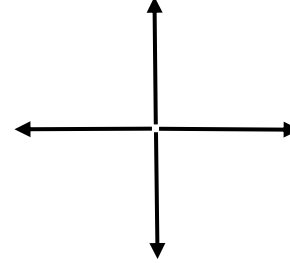
Koşulların kontrol
edildiği ve koşullara
göre verilen kararları
yönlendirmek için
kullanılır.



Yazdırma
sembolü



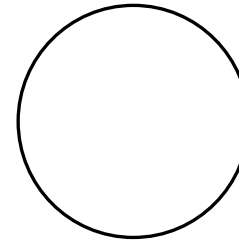
Bir sonucun ya da bir
raporun yazdırıldığını
gösterir.



Akış yönlerini
gösterir.



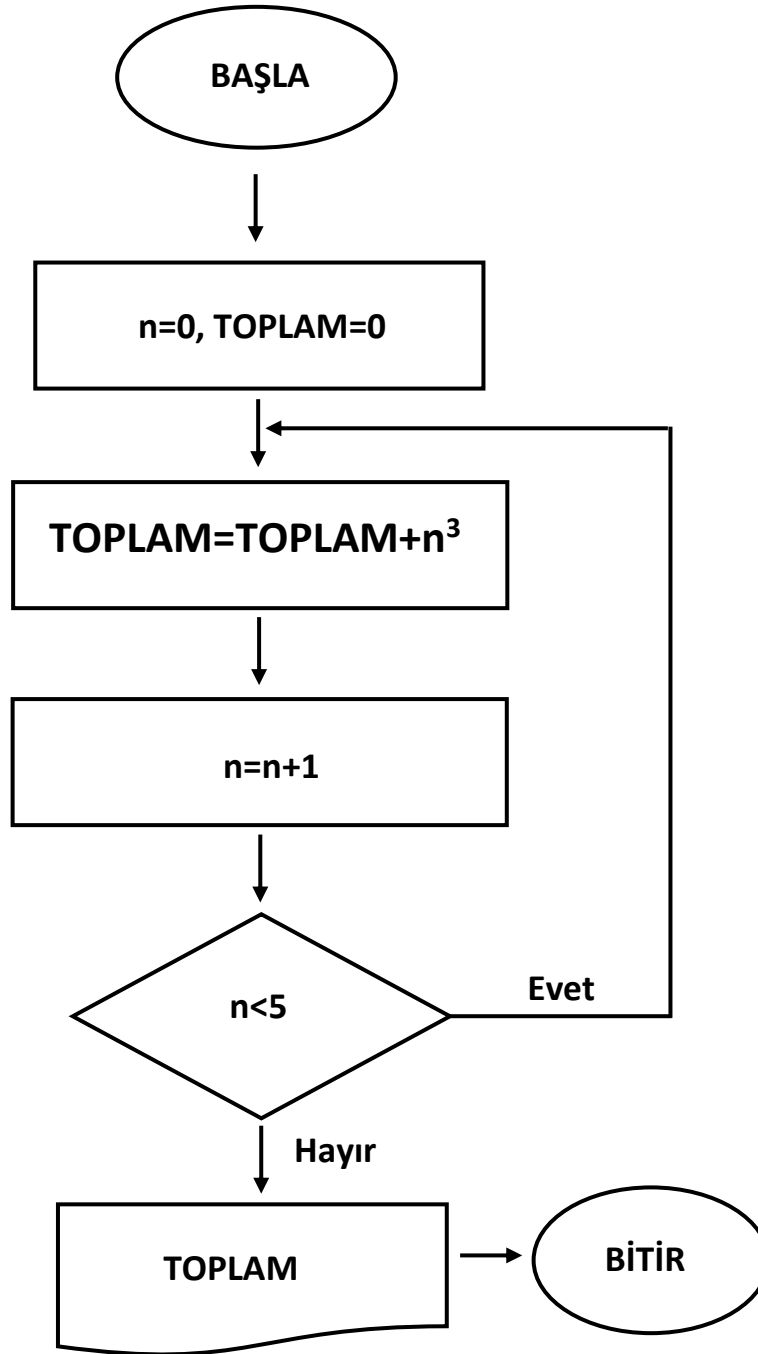
İstenen şartlarda
tekrar eden işlemler
için kullanılır.



Birleştirici veya
bağlantı noktalarını
temsil eder. Akışın
başka bir yere
bağlandığını gösterir.

Ö:

Yanda
verilen
işleyişin
sonucu
(ekran
görüntüsü)
kaç olur?



Buradaki toplam değeri kumbaraya atılan para gibi düşünebiliriz. İlk başta kumbarada değer yok. Yani sıfır. Her bir değer yani para atıldıkça kumbaranın içindeki para ile toplanacak. Buradaki n değeri parayı temsil ediyor.

Durum	n değeri	n^3	toplam
0.Durum	0	0	0
1.Durum	1	1	1
2.Durum	2	8	9
3.Durum	3	27	36
4.Durum	4	64	100
5.Durum	5	Döngü biter	

Döngü $n<5$ ifadesi Hayır dır.

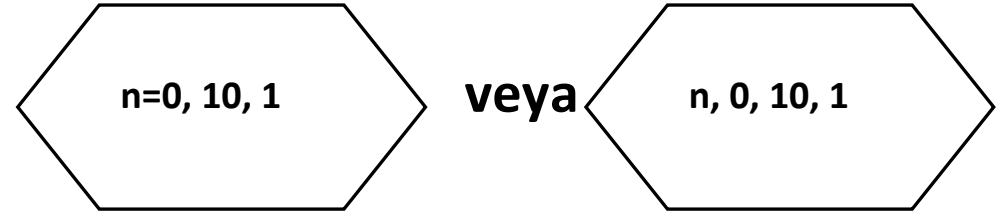
Not: Döngü için şematik olarak ALTIGEN kullandığımızı daha önce söylemiştik. Döngü Başlangıç bitiş ve artış değeri olan ritmik sayma işlemini algoritmik olarak bilgisayara iletmektir. Nasıl ki öğretmeni öğrencisine 3 ten başla 5'er 5'er 100 e kadar say diyorsa aynı işlemi bilgisayarda yapmak mümkündür.

Ör:



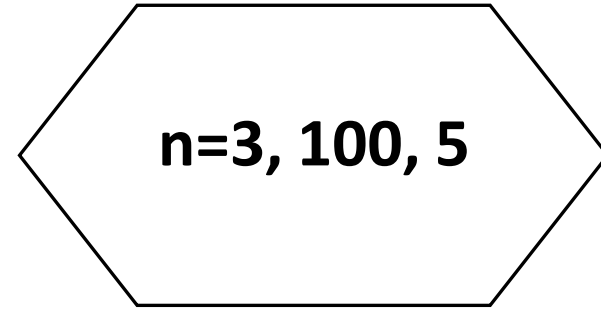
Buradaki tanımlamalar sırasıyla

- **n: döngü değişkeni**
- **a: döngü başlangıç değeri**
- **k: döngü bitiş değeri**
- **b: döngü artış veya azalış değeri**



Anlamı: Başlangıç değeri 0, artış miktarı 1 olan ve 10 kez tekrar edecek olan bir işlemi gösterir.

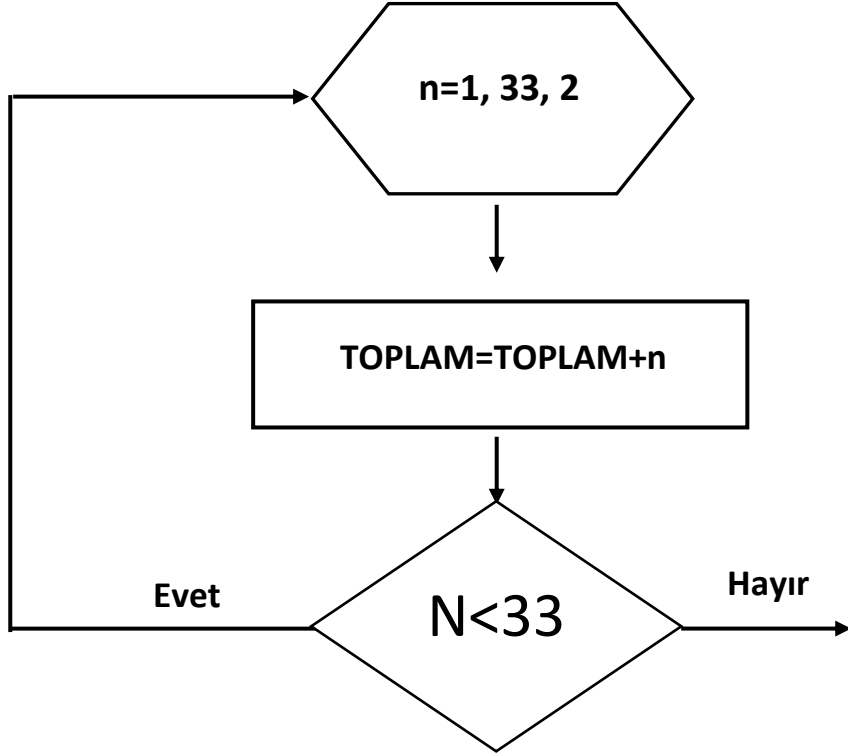
Veya öğretmenin öğrenciye söylediği



Sonuc: 3,8,13,18.....98

Sayma işlemi 98 de durur. Çünkü bir sonraki işlemde 103 olacaktır. Bizim sınırlamamız 100 olarak belirlendi.

Ö: 1 ile 33 arasındaki tek sayıların toplamını bulan döngünün akış şeması aşağıda verilmiştir



Başlangıç yani n=1 (dahil)

Bitiş: 33 (dahil değil)

Artış: 2

Bu şemanın ekran görüntüsü aşağıdaki şekilde olur.

Durum	n değeri	Toplam
1. Durum	1	1
2. Durum	3	4
3. Durum	5	9
4. Durum	7	16
5. Durum	9	25
6. Durum	11	36
7. Durum	13	49
8. Durum	15	64
9. Durum	17	81
10. Durum	19	100
11. Durum	21	121
12. Durum	23	144
13. Durum	25	169
14. Durum	27	196
15. Durum	29	225
16. Durum	31	256
17. Durum	33	Döngü Biter

SON OLARAK !!!

Not: Döngü, algoritmada bir işlemin belirli bir şart sağlandığı sürece tekrar tekrar yapılmasıdır.

Yani aynı işi defalarca yazmak yerine, bilgisayara “bunu tekrar et” demektir.

Döngü Neden Kullanılır?

- Tekrar eden işleri kolaylaştırmak için
- Kod yazmayı kısaltmak için
- Hata yapma ihtimalini azaltmak için

Son Soru: 100'den başlayarak 8'er azaltarak negatife düşmeden sayıları ekrana yazdırmak. Algoritma olarak yazacak olursak.

- 1.Adım: Başla
- 2.Adım: 100'ü ekrana yaz
- 3.Adım: $100 - 8 = 92 \rightarrow$ ekrana yaz
- 4.Adım: $92 - 8 = 84 \rightarrow$ ekrana yaz
- 5.Adım: $84 - 8 = 76 \rightarrow$ ekrana yaz
- 6.Adım: $76 - 8 = 68 \rightarrow$ ekrana yaz
- 7.Adım: $68 - 8 = 60 \rightarrow$ ekrana yaz
- 8.Adım: $60 - 8 = 52 \rightarrow$ ekrana yaz
- 9.Adım: $52 - 8 = 44 \rightarrow$ ekrana yaz
10. Adım: $44 - 8 = 36 \rightarrow$ ekrana yaz
11. Adım: $36 - 8 = 28 \rightarrow$ ekrana yaz
12. Adım: $28 - 8 = 20 \rightarrow$ ekrana yaz
13. Adım: $20 - 8 = 12 \rightarrow$ ekrana yaz
14. Adım: $12 - 8 = 4 \rightarrow$ ekrana yaz
15. Adım: Bitir

**Burada dikkat ederseniz döngü yok.
Şimdi aynı soruyu döngü ile yapalım.**

- 1.Adım: Başla
- 2.Adım: $sayı = 100$ yap
- 3.Adım: Eğer $0 \geq sayı$ ise git 6. adıma
- 4.Adım: $sayı$ 'yı ekrana yazdır
- 5.Adım: $sayı = sayı - 8$ git 3. adıma
- 6.Adım: Bitir.

Az önceki algoritmada yazılan sözde kodlar 15 adım sürmüştü. Oysaki burada döngü sayesinde sadece 6 adımda bitti.

Buradaki sayının başlangıç bitiş azalış veya artış miktarı ne olursa olsun hep 6 adımda bitecekti. Şimdi aynı işlemi akış şeması olarak yapalım.