

ALGORİTMA

Günlük hayatta birçok etkinliği daha verimli hale getirmek için sistematik kurallar ve adımlar izlenebilir. Örneğin yoğurt yapmak, bir sayının başka bir sayıyla kalanını bulmak gibi.

Algoritma kelimesi ünlü matematikçi Harizmi'nin isminin Avrupalılar tarafından yanlış telaffuz edilmesi sonucu ortaya çıkmıştır.

Örnek...1 :

1. adım: Aklından bir sayı tut.
2. adım: Tuttuğun sayının karesini al.
3. adım: 2. adımda elde ettiğin sayıdan 9 çıkar.
4. adım: 3. adımda elde ettiğin sayıyı aklından tuttuğun sayının 3 eksiğine böl.
5. adım: Bulduğunda cevabı yüksek sesle söyle.

Söylenen sayıya göre akıldan tutulan sayı bilinebilir mi? Açıklayınız.

Örnek...2 :

Pul oyunu

İki oyuncuyla oynanan bir pul çekme oyununda her oyuncu kendi sırası geldiğinde seçtiği sıradan en az 1, en fazla 3 rastgele pul çekebilir. Son pulu çeken oyunu kazanır.



Bu kurala göre öğrencilerden 1. oyuncunun en alt sıradan 1 pul çekmesi durumunda 2. oyuncunun oyunu kesin olarak kazanabileceği yola ilişkin bir algoritma yazınız.

ALGORİTMANIN GÖSTERİM YÖNTEMLERİ

Algoritmaların tanımlanmasında ve gösteriminde kullanılan farklı yöntemler

1. Algoritmik Doğal Dil

(Konuşma dili ile gösterim) Algoritmanın işleyişini gösteren ve bir problemi çözmeye adımlarını günlük konuşma dili kullanarak sıralı ve açık biçimde herkesin anlayabileceği temel cümlelerle anlatan bir yöntemdir.

Algoritmik doğal dil ile gösterimde standart yöntem yoktur, algoritma yazma kurallarına uyularak farklı biçimlerde yazılabilir.

Algoritmik doğal dil ile yazılırken genel adımlar **Başla—Girdilerin alınması—Hesaplama/Koşul Kontrolü— Sonucun Yazılması—Bitir** biçiminde olabilir

Örnek...3 :

Haldun'un, sabah uyandığında yapacağı kahvaltı için bir kahvaltı hazırlama algoritması oluşturalım.

- Adım 1: **Başla**
 Adım 2: Yataktan kalk
 Adım 3: Mutfaka git
 Adım 4: Ekmek al
 Adım 5: Çayı hazırla
 Adım 6: Dolaptan kahvaltılıkçı çıkar
 Adım 7: Bardak bitince çayını doldur
 Adım 8: Karnın doyduğunda sofradan kalk
 Adım 9: Sofrayı temizle
 Adım 10: **Bitir**

Bu algoritmanın adımları kısaltılıp uzatılabilir. Aynı zamanda farklı şekillerde de ifade edilebilir.

MANTIKSAL İŞLEMLER

Algoritma temelli işlemlerde mantıksal işlemler cebirsel olarak ifade edilen dilden farklılık göstermektedir. Operatörler, değişkenler veya sabitler üzerinde matematiksel ve karşılaştırma işlemlerini yapan simgelerdir.

Aritmetik Operatörler

+	Toplama
-	Çıkarma
x	Çarpma
/	Bölme
%	Bölmede kalanı bul

Karşılaştırmalı Operatörler

>	Büyüktür
<	Küçüktür
==	Eşittir
>=	Büyük – Eşittir
<=	Küçük – Eşittir
!=	Eşit değil

Örnek...4 :

İki doğal sayının toplama işlemini yapan algoritmanın işleyişini algoritmik doğal dil ile verilmiştir, inceleyiniz.
Doğal sayılar m ve n olsun.

Adım 1: **Başla**

Adım 2: **Girdilerin alınması**

m doğal sayısını al

n doğal sayısını al

Adım 3: **Toplamın hesaplanması**

Toplam = m + n işlemini yap

Adım 4: **Sonucun Yazılması**

Toplam değerini ekrana ya da bir dosyaya yazdır

Adım 5: **Bitir**

Örnek...5 :

Kenar uzunlukları verilen—girilen (p ve r olsun) bir dikdörtgenin alanını bulan algoritmanın işleyişini algoritmik doğal dil ile oluşturulmuştur. Boşlukları doldurunuz

Adım 1: **Başla**

Adım 2: **Girdilerin alınması**

.....

Adım 3: **Alanın hesaplanması**

.....

Adım 4: **Sonucun yazdırılması**

.....

Adım 6: **Bitir**

Örnek...6 :

Adım 1: **Başla**

Adım 2: x, y, z, k gir

Adım 3: $p = (x + y + z)/4$

Adım 4: ortalama p olarak elde edilir.

Adım 5: **Bitir**

a) Yukarıda verilen algoritmanın işlevini açıklayınız.

b) Yukarıdaki algoritmanın işleyişinde 25, 36, 47 ve 40 sayıları klavyeden girildiğinde elde edilen sonucu bulunuz.

Örnek...7 :

Bir ürünün fiyatını, %18 KDV ekleyerek ekrana yazdıran algoritmanın işleyişini algoritmik doğal dil ile oluşturunuz

Adım1

Adım2

Adım3




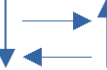


Adım4

Adım5

2. Akış şeması ile gösterim

Akış şeması, bir problemi çözme adımlarının görsel temsidir. İşlemleri, karar noktalarını (evet veya hayır) ve akış yönlerini çeşitli geometrik şekiller kullanarak gösterir.

AKIŞ ŞEMASINDA KULLANILAN SEMBOLLER VE ANLAMLARI

Sembol	Anlamı
	Akış şemasının başlangıcını veya sonunu gösterir.
	Veri girişini veya çıkışı temsil eder. Değişkenlere değer atamak için kullanılır.
	Bir koşulun kontrol edildiği karar noktasını ifade eder. Akış birden fazla yola ayrılır. Evet/Hayır veya Doğru/Yanlış şeklinde kararlar bu şekilde gösterilir.
	Akış yönlerini temsil eder.
	İşlem yapmak için kullanılır.
	Algoritma sonucunun yazdırılmasını temsil eder.

—Akış şemasının görselliği algoritmanın daha kolay anlaşılabilmesine ve takibinin kolay yapılmasına olanak sağlar

—Atama operatörleri, değişkenlere veya veri yapılarına değer atamak için kullanılan simgelerdir. Bu operatörler; bir değişkenin değerini başlatmak, güncellemek veya bir hesaplama sonucunu belirlemek için kullanılır. Atama için " ← " veya "=" sembolleri kullanılabilir.

Örnek...8 :

a: birinci sayıyı, b: ikinci sayıyı, c ise a ve b sayılarının çarpımını göstermek üzere, iki doğal sayının çarpımının bulunmasıyla ilgili algoritmanın doğal dil ile yazılmış hali aşağıda solda, akış şeması ise sağda verilmiştir. İnceleyiniz.

Algoritmik doğal dil

Adım 1: Başla

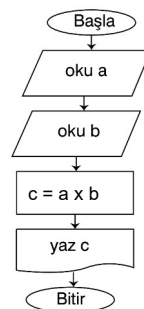
Adım 2: a'yı oku.

Adım 3: b'yi oku.

Adım 4: $c = a \times b$

Adım 5: c'yi yaz.

Adım 6: Bitir

Akış şeması

Örnek...9 :

Aşağıda klavyeden girilen a ve b sayılarının ortalamasını bulan algoritmanın işleyişine ait akış şeması verilmiştir. Boşlukları uygun şekilde doldurunuz.

Algoritmik doğal dil

Adım 1: Başla

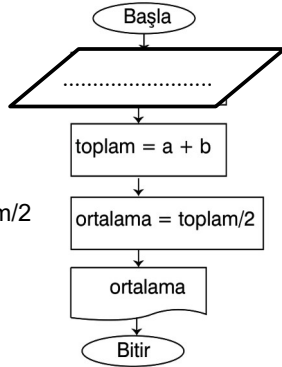
Adım 2: a'yı ve b yi oku.

Adım 3:

Adım 4: ortalama=toplam/2

Adım 5:.....

Adım 6: Bitir

Akış şeması**Eğer ise.... Değilse Bağlacının Kullanımı**

"Eğer P ise R " ve "Değil" tek bir karşılaştırma deyimidir. Burada "Değil" kullanımı şart değildir.

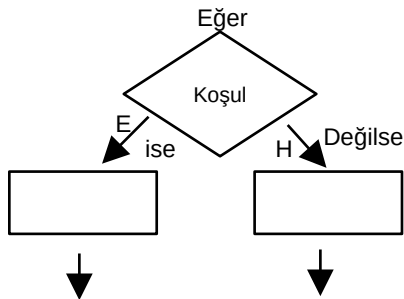
Eğer koşul gerçekleşiyorsa "Eğer" den sonraki bölüm yürütülür ve "Değil" den sonraki bölüm atlanır.

Koşul gerçekleşmiyorsa "Eğer" den sonraki ifade atlanır ve mevcutsa, "Değil" den sonraki işlemler gerçekleştirilir.

Eğer (koşul) { Doğru ise İşlem1}
Değil { Yanlış ise İşlem 2}

Örneğin; Eğer sayı % 2=0 ise yaz "girdiğiniz sayı çifttir." Değilse yaz "girdiğiniz sayı tektir."

Genel Olarak



"**karar sembolü**" mantık bağlaçlarından **ise** sembolü ile ifade edilir.

Koşulun doğru olması durumunda Evet (E) yazılı ok yönünde, yanlış olması durumunda Hayır(H) yazılı ok yönünde işleme devam edilir.

Örnek...10 :

Verilen bir sayının (p) verilen başka bir sayı (r) ile bölümünden kalanı (p>r)tekrarlı çıkarma yöntemiyle bulan algoritmanın işleyişini algoritmik doğal dil ve akış şeması ile ifade edilmiştir. Akış şemasını uygun şekilde doldurunuz.

Algoritmik doğal dil ile

1. adım: **Başla**.

2. adım: **Girdilerin alınması**
Kullanıcıdan bölünen (p)
ve bölen (r) al.

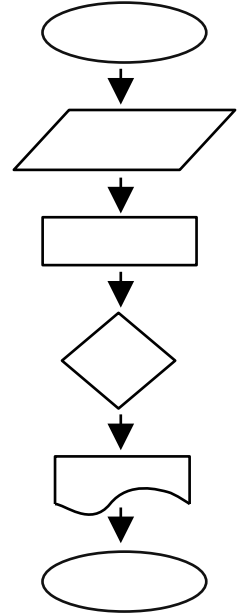
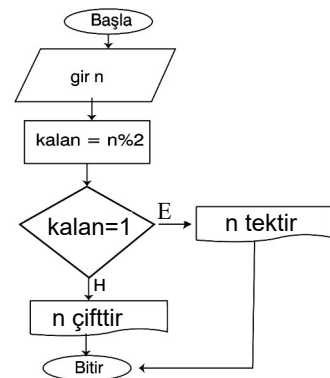
3. adım: **Kalanın güncellenmesi**
p=p-r olarak hesapla

4. adım **Koşulun kontrolü**
Adım 3 de bulunan p sayısı
r den küçükse 5. adıma,
büyükse 3.adıma git

5.adım: **Sonucun bulunması**
Kalan 3.adımda bulunan
p sayısıdır.

6. adım: **Bitir**.

Akış Şemasıyla

**Örnek...11 :**

Klavyeden girilen bir n pozitif tam sayısı için işleyişini aşağıdaki akış şemasıyla verilen bir algoritma verilmiştir. Algoritmanın işlevini açıklayınız.

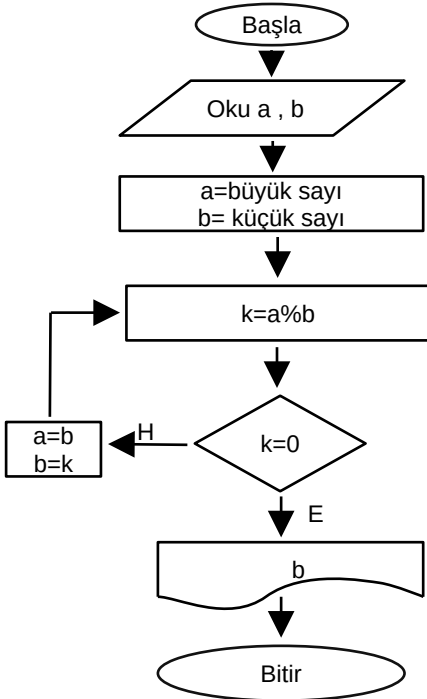
Örnek...12 :

Klavyeden girilen x gerçekte sayısının mutlak değerini ekrana yazdıran programın algoritmasını

- Algoritmik doğal dil ile ifade ediniz
- Akış şemasını yazınız.

Örnek...13 :

Aşağıda bir algoritma ile ilgili akış şeması gösterilmiştir. ($a, b \in \mathbb{Z}^+$)



Buna göre, aşağıdaki soruları cevaplayınız.

- $A = 600$ ve $B = 144$ için algoritmanın vereceği sonucu bulunuz.
- Akış şemasının işlevini açıklayınız.
- Akış şemasını algoritmik doğal dil ile yazınız.

3. Söзде kod

Söзде kod; bir bilgisayar programının nasıl çalışacağını anlatan, gerçekte bir programlama dilinin teknik detaylarını ve karmaşık söz dizimini kullanmayan, açıklayıcı bir dil kullanılarak yazılan bir dizi yönergedir.

Söзде kodda;

* **oku, yaz, göster** gibi komutlarla temel okuma ve yazma işlemleri gerçekleştirilir.

* Algoritmik adımlar yazılırken "**Başla**" ile başlanır "**Bitir**" ile bitirilir.

* Adım numaralandırma genelde yapılmaz.

Söz dizimi, bir programlama dilindeki komutların, ifadelerin ve veri yapılarının nasıl düzenlenmesi gerektiğini tanımlayan kurallar ve yapılar bütünüdür

Örnek...14 :

Bir taban uzunluğu ve o tabana ait yüksekliği girilen üçgenin alanını bulan algoritmanın işleyişi söзде kodla aşağıdaki gibi gösterilebilir. İnceleyiniz.

1.yol

Girdi: taban, yükseklik

Çıktı: alan

Başla

alan=tabanxyükseklik/2

yazdır " Üçgenin alanı ", alan, birimkaredir

Bitir

2.yol

Başla

(gir) oku a sayısı, h sayısı

alan = $axh/2$

Yaz "alan"

Bitir

Örnek...15 :

Klavyeden girilen n pozitif tam sayısının tek ya da çift olduğunu yazan algoritmaya ait söзде kod verilmiştir için , boşlukları doldurunuz.

Girdi pozitif tamsayı,

Çıktı Teklik/çiftlik durumu

Başla

Gir n sayısı

Kalan =

Eğer kalan =..... ise yaz "n, çifttir"

Değilse yaz "n,"

Bitir

Örnek...16 :

İki sayının aritmetik ortalamasını hesaplayan algoritmayı sözde kod ile yazılmıştır, boşlukları doldurunuz.

Girdi sayı1 , sayı2 .

Çıktı Ortalama

Başla

gir sayı1, sayı 2

toplam =

ortalama =.....

yaz "ortalama"

Bitir

Örnek...17 :

Üniversitede okuyan bir öğrencinin geçme puanı için vize sınavının % 40'ı final sınavının % 60' ı şeklindedir. Bu öğrenci 50 ve 50'den yüksek puan alırsa "başarılı", 50'den küçük puan alırsa "başarısız" sayılacaktır. Öğrencinin başarı durumunu veren bir sözde kod yazılmıştır, boşlukları doldurunuz.

Girdi Vize puanı (a) ve final puanı (b)
Çıktı Başarı durumu

Başla

gir vize puanı, final puanı

a = v×0,4

b =

d =+.....

Eğer d >= 50 ise yaz "başarılı"

Değilse yaz "başarısız"

Bitir

Örnek...18 :

9 yaş ve üzeri öğrenciler için gösterime giren bir korku filmi ile ilgili yaşı tutuyorsa "giriş yapınız" tutmuyorsa "giriş yasak" yazan algoritmanın işleyişini sözde kod ile yazılmıştır, boşlukları doldurunuz..

Girdi: Yaş

Çıktı: Filme girebilme durumu

Başla

göster "yaş" giriniz.

oku "yaş"

..... yaş > = 9 yaz "giriş yapınız."

..... yaz "giriş yasak"

Bitir

Örnek...19 :

Gerçek Sayılar kümesinde $f(x)=ax+b$ şeklinde tanımlı doğrusal fonksiyonun sıfırını bulan algoritmanın işleyişini sözde kodla ifade ediniz.

Örnek...20 :

N pozitif tam sayısının 5'e tam bölünüp bölünmediğini bulan bir algoritmanın işleyişini sözde kodla yazınız.

Sözde kod yazımında kullanılan yazım tarzlarında farklılıklar olabilir. (Standart bir kullanım şekli yoktur) Bu farklılıklar, kullanıcının yatkın olduğu programlama diline kolay geçiş yapmasını sağlar.

Algoritmik doğal dil, akış şeması ve sözde kod alanlarındaki ifadelerde imla ve noktalama farklılıklarının bulunma nedeni bu ifadelerin kullanımına ilişkin kuralların yazı dili kurallarından farklı olmasıdır.

Programlama dili, bilgisayarların anlayabileceği ve işleyebileceği talimatlar dizisini oluşturmak için kullanılan formel bir dildir. (Basic, Pascal, Java gibi) Her programlama dilinin kendine özgü bir söz dizimi vardır. Programcılar, söz dizimleri yardımıyla algoritmaları kod haline getirir. Böylece bilgisayarlar istenen görevleri yerine getirebilir. Söz dizimi, programlama dilinin kurallarını ve yapılarını içerir. Doğru kullanılmadığında program hatalı çalışır veya hiç çalışmaz.

ALGORİTMALARDA SAYAÇ KULLANIMI

Programlarda, bazı işlemlerin bir kereden fazla yaptırılması gerekebilir. Bu tür sayma işlemlerine algoritmada sayaç adı verilir.

Sayaç, işlem akışı kendisine geldiğinde, belirtilen adım değeri kadar artan veya azalan bir değişkendir.

$sayaç = sayaç + k$
gösteriminde değişkenin eski değerinin üzerine k eklenerek bulunan sonuç yine değişkenin kendisine yeni değer olarak atanmaktadır.

Örnek...21 :

Sayaç yapısını kullanarak 100'e kadar olan doğal sayılardan beşin katlarını ekrana yazdıran programın algoritmasının algoritmik doğal dil ile ifadesi aşağıda verilmiştir.

- Adım 1: Başla
Adım 2: sayaç = 0
Adım 3: yaz sayaç
Adım 4: sayaç = sayaç + 5
Adım 5: Eğer sayaç <= 100 ise Git Adım 3
Değilse Adım 6'ya git.
Adım 6: Bitir

Yukarıdaki verilen algoritmanın işleyişini açıklayınız.

ERATOSTHENES KALBURU

Eratosthenes (Eratostenis) Kalburu, antik çağda Yunan matematikçisi Eratostenes tarafından geliştirilen bir algoritmadır. Bu algoritma ile, bir n doğal sayısından küçük olan tüm asal sayıları bulmak için kullanılır ve bir sayının daha küçük asal sayıların katı olup olmadığını kontrol ederek asallığını test eder. (Yöntemde bir sayısını çizdikten sonra iki ile başlayıp asal sayılarının katlarını, araştırılacak sayıya kadar listeden çizerek çıkarılır). Aşağıda algoritmik doğal dille yöntem yazılmıştır. İnceleyiniz.

X	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

- Adım 1: Başla
Adım 2: Kullanıcıdan n sayısını al
Adım 3: i=2
Adım 4: Eğer i <= n ise adım5 e değilse adım 7 ya git
Adım 5: Eğer i işaretli ise adım6 ya değilse i yi asal olarak daire içine al ve i nin n den büyük olmayan katlarını işaretle
Adım 6: i=i+1 adım 4 e git
Adım 7: Bitir

ALGORİTMİK YAPILAR İÇERİSİNDEKİ MANTIK BAĞLAÇLARI VE NİCELEYİCİLER

Algoritmik yapılar içinde kullanılan mantık bağlaçlarının işlevleri (mantıksal operatörler) aşağıda verilmiştir.

Operatörler
Ve \wedge
Veya \vee
İse \Rightarrow
Ya da \vee
Her \forall
Bazı \exists

Ve bağlacı: İki veya daha fazla önermenin hepsinin doğru olup olmadığını kontrol eder.

Veya bağlacı: İki veya daha fazla önermeden en az birinin doğru olup olmadığını kontrol eder.

Ya da bağlacı: İki önermeden yalnızca birinin doğru olup olmadığını kontrol eder.

İse bağlacı: Bir koşulun sağlanması durumunda belirli bir komutun yerine getirilmesini sağlar.

Her Niceleyicisi Bir önermenin herhangi bir öge için doğru olduğunu ifade eder.

Bazı Niceleyicisi Bir önermenin en az bir öge için doğru olduğunu ifade eder.

Mantık Bağlaçları ve Niceleyicilerin Matematiksel İspat ve Algoritmalarındaki İşlevleri

△ Koşullara Göre Karar Verme
Mantık bağlaçları, ispat süreçlerinde karar noktalarını belirlemek ve farklı durumlar arasında geçişi kontrol etmek için kullanılır.

△ **Adımları Birleştirme**
Mantık bağlaçları, bir ispatın adımlarını birbirine bağlamak ve sonuçların doğruluğunu sağlamak için kullanılır.

△ **Genelleme ve Özelleştirme**
Niceleyiciler, matematiksel ifadelerde genelleme veya özelleştirme yapmak için kullanılır.

△ **Sembolik İfadelerin Anlaşılması**
Niceleyiciler ve mantık bağlaçları, sembolik ifadelerin doğru yorumlanmasını sağlayarak matematiksel dilin anlaşılabilirliğini ve netliğini artırır.

Örnek...22 :

Aşağıda bir oto galeride satışta bulunan araçlara ait özellikler verilmiştir. Buna göre soruları cevaplayınız

Adı	Yaşı	Motor Hacmi (lt)	Güvenlik (*)	Fiyatı (bin ₺)
A	3	1000	3	750
B	4	1200	4	900
C	2	1600	5	1250
D	5	2000	4	3000
E	7	3500	5	2500

a) Yaşı 4den büyük ve güvenlik en az 4 yıldız alan araçlar hangileridir.

b) Motor hacmi 1600lt den büyük veya 1 milyon ₺ altında fiyatlı olan araçlar hangileridir?

Örnek...23 :

Vücut kitle indeksinde (vki) 25,0 ten 29,9 a kadar olan değerler için bir kişinin genel sağlık durumu "fazla kilolu" olarak kabul edilir. Bu durumu gösteren ifadenin sözde kod içerisindeki yazıldığı satırı nasıl olmalıdır ?

Örnek...24 :

İki basamaklı bir doğal sayının

a) 12 ile tam bölünüp bölünmediğini gösteren bir önerme yazıp doğruluğunu ispatlayınız .

b) İspat adımlarını kullanarak sayının 12 ye tam bölünüp bölünmediğini bulan bir algoritmanın işleyişini algoritmik doğal dil, sözde kod ve akış şemasıyla yazınız.