

Algoritmalar

Bir bilgisayar programı aslında sıra düzensel olarak tanımlanmış bir dizi komuttan başka bir şey değildir. Bu açıdan bizim yazmaya çalışacağımız programda bir dizi komut yani eylem topluluğudur. Her programda bu eylemler yazıldıkları sırada gerçekleştirilir veya çalıştırılırlar. Aslında bizim günlük hayattaki yaşantı tarzımız dahi düzenli olarak bir takım işlemlerin sıra ile yapılması şeklindedir. Yani bir iş yapabilmek için bir takım alt iş veya olayları peş peşe gerçekleştiririz.

Algoritmanın tanımını daha önce vermiştik burada bu tanımları tekrar etmek faydalı olacaktır. **Bir sorunu çözebilmek için gerekli olan sıralı mantıksal adımların tümüne algoritma** denir. Bir algoritmadan beklenen bir takım özellikler olduğunu da yine daha önceki tanımlar bölümünde bahsetmiştik. Biz şimdi mümkün olduğu kadar bu tanım ve özelliklerden yola çıkarak örneklerle bir kaç algoritma vermeye çalışacağız.

Öncelikle bir ev hanımının pasta yapmak istediğini varsayalım. Bu pastanın yapılabilmesi için gerekli bir takım işlemler ve alt adımlar bellidir. bir ev hanımı da sıra ile bu adımları uygulayarak bu pastayı yapar. Şöyle ki:

1. Pastanın yapımı için gerekli malzemeleri hazırla
2. Yağı bir kaba koy
3. Şekeri aynı kaba yağın üzerine koy
4. Yağ ve şekeri çırp
5. Karışımın üzerine yumurtayı kır
6. Tekrar çırp
7. Kıvama geldi mi diye kontrol et
8. a. Kıvamlı ise 9. adıma devam et
 b. Değilse 6. adıma dön.
9. Karışımına un koy
10. Karışımına vanilya, kabartma tozu vb. koy
11. Karışımı Kıvama gelinceye kadar çırp
12. Pastayı Kek kalıbına koy
13. Yeteri kadar ısınan fırına pastayı koy
14. Pişimi diye kontrol et
15. a. Pişmiş ise 16. adıma devam et
 b. Değilse 14. adıma dön
16. Keki fırından çıkart
17. Fırını kapat
18. Keki ı kapat
19. Keki soğumasını bekle
20. Keki servis edebilirsin.

Bu algoritma günlük hayattan bir örnek. Gerçekte biz her işimizi algoritmik olarak yaparız ancak bunu farkına varmayız. Yukarıdaki algoritmayı inceleyecek olursak bir kekin yapılması için gerekli tüm adımlar sıra ile yer almış durumda. Gerçi algoritma anlatacağımız konuların daha iyi anlaşılabilmesi için biraz farklı ele alınmıştır ama gerçek bir Pasta yapım aşamasını içerir. Bu algoritma ve diğer tüm algoritmalar için bilmemiz gereken bazı konular bulunmaktadır:

Her adım son derece belirleyici olmalıdır. Hiç bir şey şansa bağlı olmamalıdır.

Belirli bir sayıda adım sonunda algoritma sonlanmalıdır.

Algoritmalar karşılaşılabilecek tüm ihtimalleri ele alabilecek kadar genel olmalıdır.

Algoritmada algoritmanın genel işleyişini etkileyebilecek hiç bir belirsizlik olmamalıdır. (Bu örnekte öyle bir belirsizlik var. Bir fırının yeteri kadar ısınması hangi koşula bağlıdır, bu fırın ne zaman açılmış olmalıdır ve kaç dereceye ayarlanmış olmalıdır. gibi...)

Algoritmada bazı adımlar yer değiştirebilir . Ancak bir çok adımın kesinlikle yer değiştiremeyeceğini bilmeliyiz. Yanlış sıradaki adımlar algoritmanın yanlış çalışmasına neden olacaktır. (9 ve 10. adım değiştirilebilir. 2-3. adımlar yer adımlar yer değiştirebilir.) Ancak 13-16. adımlar kesinlikle yer değiştiremezler.

Peki Bilgisayarda çözülecek bir sorunu nasıl algoritma ile ifade ederiz? Bunun için öncelikle bir sorun tanımlayalım. Başlangıçta basit olması için şöyle bir problem üzerinde düşünelim: *Bilgisayara verilecek iki sayıyı toplayıp sonucu ekrana yazacak bir program için algoritma geliştirmek isteyelim.* Sorun son derece basit ancak sistem tasarımının net yapılabilmesi için sorun hakkında anlaşılamayan tüm belirsiz noktalar açıklığa kavuşturulmalıdır. Örneğin sayılar bilgisayara nereden verilecek, Klavye, Dosya veya belki başka bir ortam. Bu ve buna benzer soru ve tereddütleriniz varsa sorun sahibine bunları sormalı ve sistem analizi yapmalısınız.

Sonra bulacağımız çözümü algoritma haline dönüştürebiliriz.

1. BAŞLA
2. A sayısını oku
3. B sayısını oku
4. TOPLAM=A + B işlemini yap
5. TOPLAM değerini ekrana yaz
6. SON

Biraz daha karmaşık bir sorun şöyle olsun: *Klavyeden girilecek iki sayıdan büyük olanından küçük olanını çıkarıp sonucu ekrana yazacak program için bir algoritma geliştiriniz.*

- A. BAŞLA
- B. A sayısını oku
- C. B sayısını oku
- D. Eğer A büyüktür B $SONUC=A-B$
Değilse $SONUC=B-A$
- E. SONUC değerini ekrana yaz
- F. SON

Gerçekte bir algoritma genellikle üç ana bölümden oluşur. Bunlar :

- Algoritmanın giriş ve ilk işlemlerinin yapıldığı bölüm
- Döngüsel bir bölümün olduğu kesim. Bu bölüm iterasyonlarla bir işlemin sürekli tekrar edilerek sonuca ulaşmayı sağlayan bölümdür.
- Son işlemlerin yapıldığı bölümdür. Bu bölüm elde edilen sonuçların ekrana yazılmasını sağlayan bölümdür.

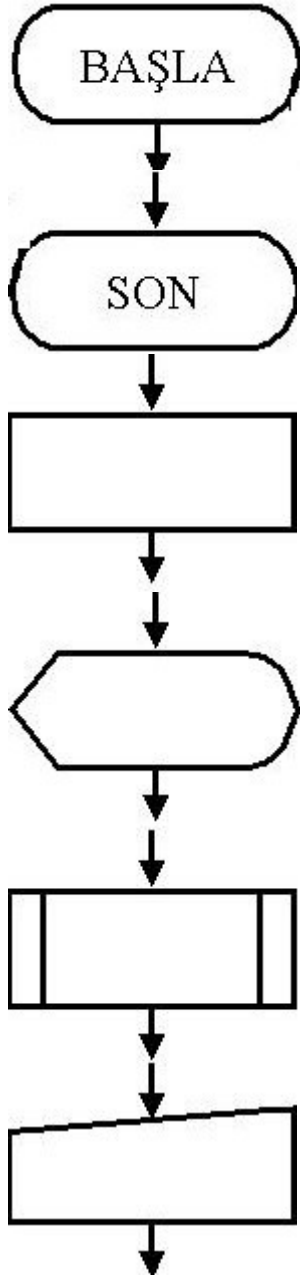
Örneğin birden Klavyeden girilen bir n değerine kadar sayıları toplayan ve sonucu ekrana yazan bir algoritmayı geliştirelim.

1. BAŞLA
2. N OKU
3. $T=0$
4. $X=1$
5. $T=T+X$
6. $X=X+1$
7. EĞER $X \leq N$ İSE 5. ADIMA GİT
8. T YAZ

Bu algoritmalar oldukça basit algoritmalar olup algoritma kavramının yerleşmesini sağlayan örneklerdir.

Akış Çizgeleri

Bir algoritmanın şekillerle görsel gösterimidir. Dikkat edildiyse algoritma doğal dille yazıldığı için herkes tarafından anlaşılamayabilir ya da istenmese de başka anlamlar çıkarılabilir. Ancak akış çizgelerinde her bir şekil standart bir anlam taşıdığı için farklı yorumlanıp anlaşılamaması mümkün değildir. Bir algoritmanın ifade edilebilmesi için kullanılan şekiller ve anlamları şunlardır:



Bir algoritmanın başladığı konumu göstermektedir. Tek çıkışlı bir şekildir.

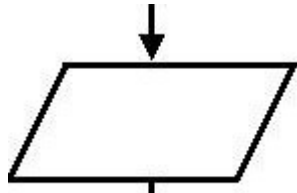
Bir algoritmanın bittiği konumu göstermektedir. Tek girişli bir şekildir.

Bir algoritmada aritmetik işlem yapılmasını sağlayan şekildir. Bu dörtgen kutu içerisine yapılmak istenen işlem yazılır. Tek girişli ve tek çıkışlı bir şekildir.

Algoritmada bir bilginin ekrana yazılacağı konumu gösteren şekildir. Ekrana yazılacak ifade ya da değişken bu şekil içerisine yazılır.

Bir algoritmada başka bir yerde tanımlanmış bloğun yerleştiği konumu gösteren şekildir. Kutu içerisine bloğun adı yazılabilir.

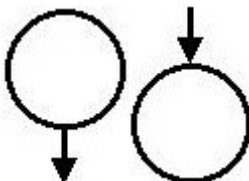
Klavyeden Bilgisayara bilgi girilecek konumu belirten şekildir. Girilecek bilginin hangi değişkene okunacağını kutu içerisine yazabilirsiniz.



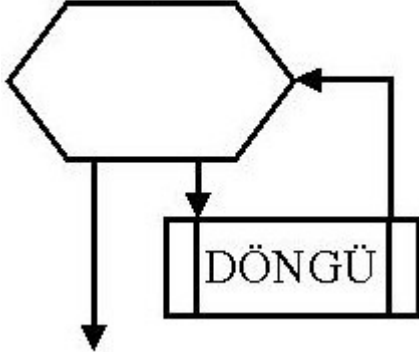
Giriş Çıkış komutunun kullanılacağı yeri belirler ve kutu içerisine hangi değişkeni ve OKU mu YAZ mı yapılacağını belirtmeniz gerekir



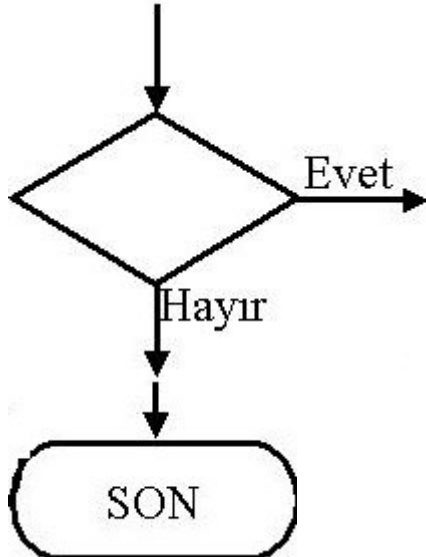
Bilginin Yazıcıya yazılacağı konumu gösteren şekildir.



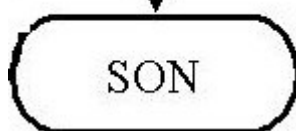
Bir algoritmanın birden fazla alana yayılması durumunda bağlantı noktalarını gösteren şekildir. Tek girişli veya tek çıkışlı olarak kullanılırlar.



Bir işlemin belli bir sayıda veya belli bir koşul doğru olduğu sürece tekrar edilmesini sağlayan döngü komutunu gösteren şekildir. Bu döngüde altıgen içerisine ya koşul ya da döngünün başlangıç, adım ve sonlanma değerlerini belirtebilirsiniz. DÖNGÜ olarak belirlenen blokta da tekrar edilmek istenen komutlar yer almaktadır.



Bir algoritmada bir kararın verilmesini ve bu karara göre iki seçenekten birinin uygulanmasını sağlayan şekildir. burada eşkenar dörtgen içerisine kontrol edilecek mantıksal koşul yazılır. Program akışı sırasında koşulun doğru olması durumunda "Evet" yazılan kısma Yanlış olması durumunda "Hayır" yazılan kısma sapılır. Tek girişli ve çift çıkışlı bir şekildir.



Programın bittiği yer ya da yerleri gösteren bir şekildir.

BAŞLA

Programlamaya Giriş ve Algoritmalar Ders Notları

N

Bu şekiller kullanılarak algoritma ile oluşturulan çözümler akış çizgelerine çevrilir. Bu şekiller tarafından anlaşılabilir ve doğru olarak yorumlanabilirler.

X=1

T=0

T=T+X

X=X+1

X<=N

E

H

T

SON

BAŞLA

A=1

B=1

C=A+B

C>100

E

SON

C

A=B

B=C

Bu örnekte 1'den N değerine kadar sayıları toplayıp ekrana yazan bir akış çizgesi çizilmiştir.

100'e kadar sayılardan fibonecci dizisinin elemanlarını listeleyen akış çizgesidir.