

Problem nedir?

Günlük hayatımızda karşılaştığımız, çözüm aranması gereken ve çözümü için bilgi, mantık, deneyim ya da dikkat isteyen durumları **problem** olarak nitelendirebiliriz.

Problemi Çözme Adımları

- Problemi anlama
- Bir plan yapma
- Planı uygulama
- Çözümü değerlendirme

Problemi anlama

Bir problemi doğru bir şekilde çözebilmenin ilk aşaması problemi anlamakla başlar, problemi anlamak aşağıda karşımıza gelecek sorulara yanıt verilebildiği zaman gerçekleşmiş olur.

- Bu problem ile ilgili elimizdeki temel bilgiler nelerdir?
- Çözüme ulaşabilmek için bu temel bilgilerin dışında ihtiyaç duyduğlarımız varsa nelerdir?
- Problemden kimler, nasıl etkilenmektedir?
- Problemin şeması oluşturulabilir mi? Anlaşılabilirliği arttırmak için problemi görselleştirebilir miyiz?
- Problem nasıl ortaya çıkmış? Bildiğimiz ve daha önce çözüm ürettiğimiz benzer problemler oldu mu?

Bir plan yapma

Her problemin çözümüne farklı çözüm yollarından ulaşılabilir, bunu için doğru çözüm planını tercih ederek uygulamak gerekmektedir. Bir problem için, aşağıda karşımıza gelecek çözüm planları uygulanabilir.

- Deneme yanılma yoluyla çözümü test etme, olası adımları ve tahmini sonuçları bulma
- Problemi daha basit parçalara ayırarak ilerleme
- Sondan başa doğru ilerleyerek farklı bir bakış açısıyla çözüme ulaşmaya çalışmak
- Olası çözümleri içeren bir görsel harita ya da zihin haritası çıkarmak (Çünkü bazılarımız çizerek çalışmayı sever)
- Beyin fırtınası yapmak

Planı uygulama

Bir planın uygulanmasının ardından sonucu başarısızlıkla sonuçlanabilir. Başarısızlıkla sonuçlanan denemeler sadece öğrenme deneyiminin bir parçasıdır.

Çözümü değerlendirme

Çözüm bulunduğunda, çözümün en etkili çözüm olup olmadığı ve bu çözümün başka problemlere uyarlanıp uyarlanamayacağı çözümü değerlendirme aşamasıdır.

Algoritma Nedir?

Bilgisayarda uygulama geliştirirken, kod yazarken, program yazarken yapacağımız işlemlerin sözel olarak, düzenli ve sıralı şekilde ifade edilmesine **algoritma** denir.

Algoritma Örnekleri

“Verilen 2 sayının toplamını bulan ve ekrana yazdıran programın algoritması”

- **Adım 1** Başla,
- **Adım 2** Birinci sayıyı gir,
- **Adım 3** İkinci sayıyı gir,
- **Adım 4** İki sayıyı topla,
- **Adım 5** sonucu ekrana yazdır,
- **Adım 6** Bitir.

“3 Öğrencinin yaş ortalamasını bulan ve ekrana yazdıran programın algoritması”

- **Adım 1** Başla,
- **Adım 2** Birinci öğrencinin yaşını gir,
- **Adım 3** İkinci öğrencinin yaşını gir,
- **Adım 4** Üçüncü öğrencinin yaşını gir,
- **Adım 5** Girilen sayıları toplayarak 3’e böl,
- **Adım 6** Sonucu ekrana yazdır,
- **Adım 7** Bitir.

“İki sayının farkını bulan ve ekrana yazdıran programın algoritması”

- **Adım 1** Başla,
- **Adım 2** Birinci sayıyı gir,
- **Adım 3** İkinci sayıyı gir,
- **Adım 4** İki sayıyı çıkar,
- **Adım 5** sonucu ekrana yazdır,
- **Adım 6** Bitir.

Tüm algoritmalar;

- **Başla** ile başlar,
- **Bitir** ile biter,
- Başla ve Bitir arasındaki işlemler **sırasıyla** gerçekleşir. İşlem basamaklarında değişiklik yapılması bize hatalı sonuç verecek ya da sonuca ulaştırmayacaktır.

AKIŞ ŞEMALARI

Algoritmaların görsel olarak **sembol** ya da **simgelerle** gösterilmesidir.

Algoritmalar sözlü ifade edilirken akış şemalarında **sembol** ya da **simgeler** kullanılır.

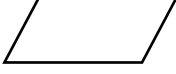
Akış Şemalarında Kullanılan Şekiller

Başla ve Bitir



Algoritmada olduğu gibi akış şemaları da ‘Başla’ ile başlar ‘Bitir’ ile biter. İkisi de aynı şekli kullanır.

Giriş işlemleri



Sayıyı giriniz, yaşınızı giriniz gibi...

Hesaplama ve

Değişkene Değer Atama

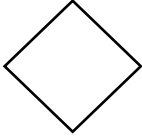


İki sayıyı topla, toplamları böl, ortalama al gibi...

Karşılaştırma

Karar Verme

(Eğer)



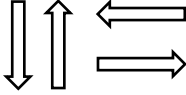
Eğer şartı uygulanır ve karar verilir. Yaşı 6’dan büyük ise, Ankara’da oturuyor ise, sınıf mevcudu 25 ise gibi

Ekran / Yazıcı Çıktı



Algoritmanın sonucunu (ekranda) bir çıktı olarak göstermek istediğimizde dalgalı dörtgen sembolünü kullanırız.

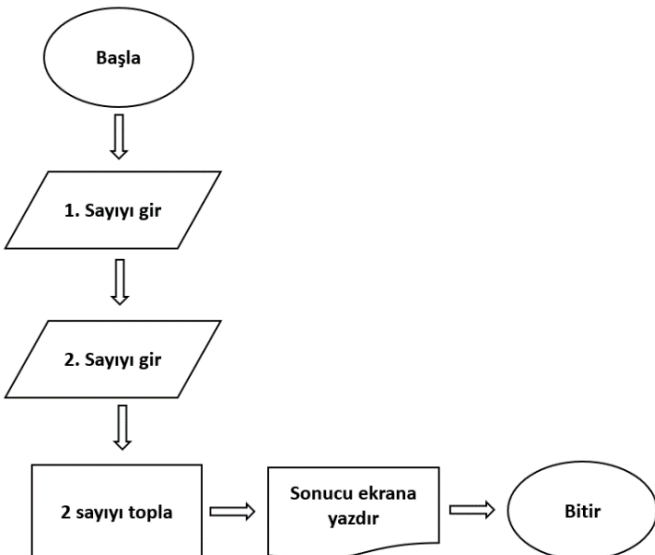
Ok İşaretleri



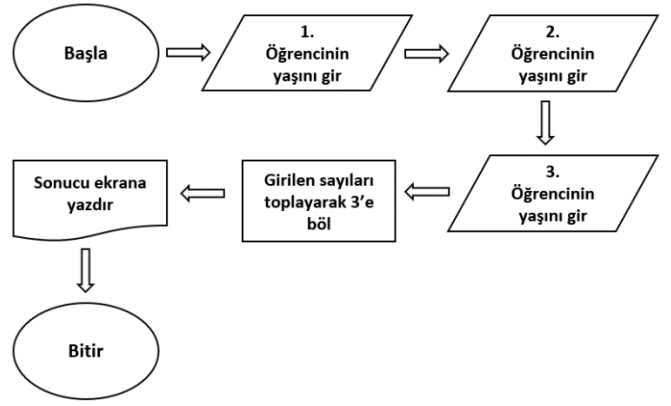
Akış diyagramındaki şekillerin birbirleri ile bağlantılarını ve işlem yönünü gösterir.

Akış Şeması Örnekleri

“Verilen 2 sayının toplamı”



“3 Öğrencinin yaş ortalaması”



Karşılaştırma ve Karar Verme İşlemlerinin Akış Şemaları

Girilen değerler ya da hesaplanan değerlere göre **şartlı işlemler** yaptırabiliriz. Şarta uyup uymama sonucuna göre işlem yönlerini değiştirebiliriz.

ÖRNEK:

Matematik dersinde öğretmenimiz sınavda 70 ve üzerinde başarı notu alan öğrencilere “Başarılısınız”; 70’in altında başarı notu alan öğrencilere de “Daha iyi çalışmalısınız” mesajı veren bir program yapmak istiyor. Programın algoritma ve akış şemasını çıkarınız.

Algoritması

- **Adım 1** Başla,
- **Adım 2** Başarı notunu gir,
- **Adım 3** Başarı notu 70 ve üzeri ise, ekrana ‘Başarılısınız’ yaz. Adım 5’e git.
- **Adım 4** Değilse, ekrana ‘Daha iyi çalışmalısınız’ yaz.
- **Adım 5** Bitir.

Akış Şeması

