











 acetinkaya / VeriYapilari-ders5





[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) [Settings](#)










MIT license



 3 stars  0 forks  1 watching  Branches  Activity


 Tags


 Public repository


 main  1 Branch  0 Tags 

 [Go to file](#)  [Add file](#)  [Code](#) 

 acetinkaya Update README.md 9b906c5 · 3 months ago 

 Ders5-Akis-Semasi.png Rename Ders5-Akis-Semasi.png to Ders5-Akis-S... 3 months ago

 LICENSE Initial commit 3 months ago

 README.md Update README.md 3 months ago

[README](#) [MIT license](#)

Python - Veri Yapıları ve Algoritmalar

Ders 5. Algoritma Hazırlığı



Bu ders içerisinde;

- 5.1. Kullanılan Operatörlerin Tam Listesi
- 5.2. Algoritmalarda Kullanılan Tanım ve Kavramlar
- 5.3. Algoritma Hazırlama Süreci

5.1. Kullanılan Operatörlerin Tam Listesi

Operatör Türü	Operatörler	Açıklama
Aritmetik Operatörler	+, -, *, /, %, **	Toplama, çıkarma, çarpma, bölme, modül, üs alma
Karşılaştırma Operatörleri	==, !=, >, <, >=, <=	Eşittir, eşit değildir, büyüktür, küçüktür, büyük eşittir, küçük eşittir
Atama Operatörleri	=, +=, -=, *=	Atama ve işlem yaparak atama
Mantıksal Operatörler	and, or, not	Ve, Veya, Değil

5.2. Algoritmalarda Kullanılan Tanım ve Kavramlar

Kavram	Tanım	Örnek/Açıklama
Algoritma	Belirli bir problemi çözmek için sıralı adımlar kümesi	Yemek tarifi, bir algoritmanın günlük hayattaki örneğidir
Girdi (Input)	Algoritmanın çalışması için gereken başlangıç verileri	Bir toplama algoritmasında sayılar girdidir
Çıktı (Output)	Algoritmanın sonunda elde edilen sonuç	Toplama algoritmasının sonucu toplamdır
Akış Diyagramı	Algoritmanın görsel olarak gösterimi	Başlangıç, işlem ve karar bloklarını içeren bir diyagram
Adım Sayısı	Algoritmanın çözüm için gerçekleştirdiği işlem sayısı	Döngülerde iterasyon sayısı adım sayısını belirtir
Karmaşıklık (Complexity)	Algoritmanın çalışma süresi veya bellek kullanımı	Big O Notasyonu ile ifade edilir ($O(n)$, $O(\log n)$)
Big O Notasyonu	Algoritmanın performansını ifade eden matematiksel gösterim	$O(n)$: Doğrusal karmaşıklık, $O(1)$: Sabit zaman
Karar Yapıları	Algoritmada koşullara göre farklı yolları temsil eder	if, else kullanımı
Döngüler	Belirli işlemleri tekrar eden yapı	for, while döngüleri.
Doğrusal Arama	Veri içinde sıralı olarak arama yapma	Liste içinde tek tek eleman kontrolü
İkili Arama	Sıralı bir veri kümesinde bölerek arama yapma	Ortadan bölerek arama işlemi
Recursive (Özyinelemeli)	Bir fonksiyonun kendisini çağırdığı algoritma türü	Fibonacci serisinin hesaplanması
Veri Yapıları	Algoritmanın üzerinde çalıştığı verilerin düzenlenme şekli	Liste, dizi, yığın (stack), kuyruk (queue)
Heuristik	En iyi çözüme ulaşmak için kullanılan tahmini veya sezgisel yöntemler	Yapay zeka problemlerinde kullanılır

5.3. Algoritma Hazırlama Süreci

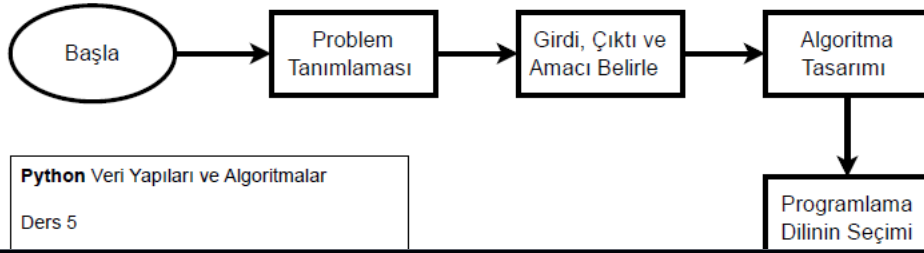
1. Başla
2. Problem Tanımlaması
3. Girdi, Çıktı ve Amacı Belirle
4. Algoritma Tasarımı :-> Adım adım çözüm sürecini tasarlamak. Burada kullanılan yöntemler: Pseudocode ve Akış Diyagramları'dır.
5. Programlama Dilinin Seçimi
6. Algoritmayı Kodlamak
7. Algoritma Analizi: Algoritmanın zaman ve alan karmaşıklığını analiz edin.
8. Test ve Düzeltme: Algoritmanın çeşitli test senaryolarında doğruluğunu test etmek ve gerekirse düzeltmeler yapmak.
9. Bitir

Pseudocode Örneği:

Bölüm 5.3. Algoritma Hazırlama Sürecinin Gösterimidir.

Akış Diyagramı Örneği:

Bir Probleme Algoritma Hazırlama Sürecindeki Yaklaşımın Akış Şeması



Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)