



VERİ YAPILARI

DATA STRUCTURE



Neler Öğreneceğiz

- Veri Yapılarına ilişkin genel tanımlar, (Ön Bilgi),
- Özyinelemeli ve Özyinelemesiz Algortimalarda Zaman Analizi,
- Bellek Gereksinimleri,
- Sıralama ve Arama Algoritmaları,
- Bağlı Liste ve Dizi Veri Yapıları,
- Yığın, Kuyruk Veri Yapıları,
- Ağaç Veri Yapısı,
- Prefix, Infix, Postfix kavramları,
- Heap Veri Modeli,
- Graf Veri Modeli,
- Hash Tablosu Veri Modeli, Hashing Uygulamaları
- Veri Sıkıştırma



Veri Yapıları ve Algoritmalar

- Veri yapıları ve Algoritmalar, “*Bilgisayar Bilimleri*” alanındaki en köklü ve klasik konulardan birisidir.



Veri Yapıları Dersinin Amacı

- Bilgisayar bilimlerinde **işlenecek veri miktarı** büyüdükçe veriye dayalı bazı problemler karşımıza çıkmaktadır.
 - Hafıza verimliliği (Memory efficiency) -Saklama
 - Zaman verimliliği (Time efficiency) -Erişim
- Bu problemler ile başa çıkmak için bazı yöntemler kullanılmaktadır.
 - Veri Yapıları ve Veri Modelleri, Verilen **saklanması**,
 - Sıralama Algoritmaları, Verilerin **sıralanması**,
 - Arama Algoritmaları, veriler içinde **arama yapılması**
- Bu gibi temel işlemlerin **verimli şekilde** (hafıza, performans) gerçekleştirilebilmesi için farklı **Veri Yapılarının** geliştirilmesi **ihtiyacı doğmuştur.**



Veri Yapıları Dersinin Amacı

- Tüm bu işlemleri gerçekleştirirken bazı algoritmalar kullanılır. Aynı zaman da bu algoritmalar için de bellek kullanımı gerçekleştirilir.
- Bu noktada amaç; **En az bellek kullanımı ile en hızlı çalışacak algoritmalar**dan faydalanmak veya **algoritmalar geliştirme farkındalığına sahip olmak.**



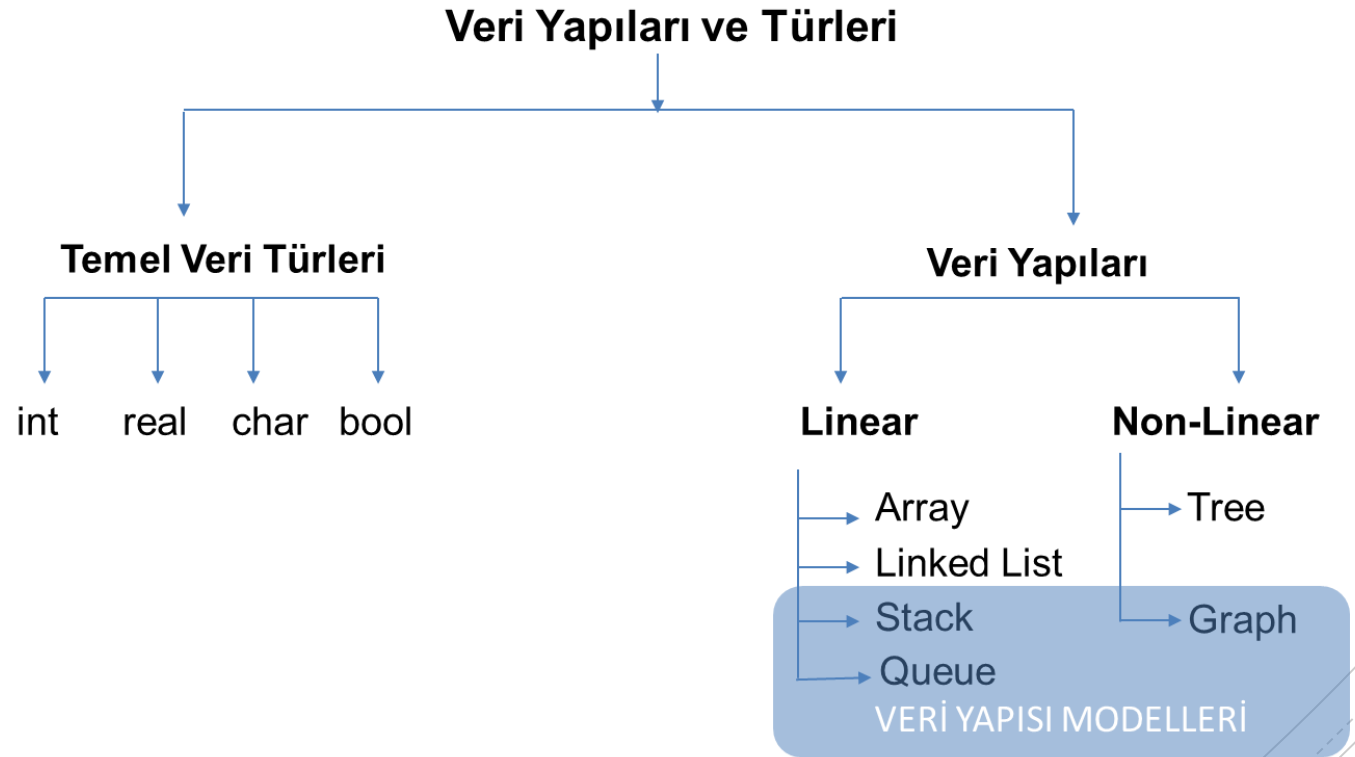
Veri Yapısı Nedir?

- Bir problemin çözümü için öncelikle *işlenecek verinin* **çözüme uygun** şekilde **organize** edilmesi gerekir. Bu bağlamda VY tanımı:
- “Veri Yapısı, verinin **verimli şekilde** kullanılabilmesi için *belirli bir formatta* **saklanıp organize edilmesini** sağlayan **matematiksel modeldir.**”



Veri Yapısı Nedir?

- Tüm programlama dillerinin, genel olarak, tamsayı, kesirli sayı, karakter ve sözcük saklanması için temel veri yapıları vardır. Bir program değişkeni bile basit bir veri yapısı olarak kabul edilebilir.



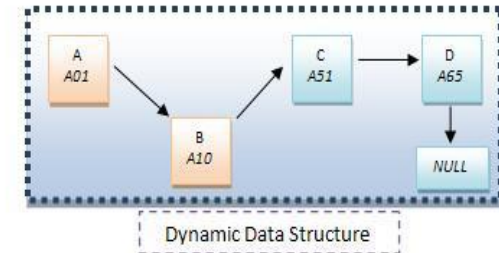
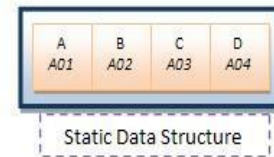
Veri Yapıları

- **Stack**, **Queue**, **Tree** ve Graph'lar **Veri Yapılarına** birer örnektir.
- Verinin *organize edilmesine* bağlı olarak **Veri Yapıları** **iki grupta** sınıflandırılırlar:
 1. **Linear VY:** Elemanların (verinin) **sırayla erişildiği** Array, Linked List, Stack ve Queue gibi yapılardır.
 2. **Non-Linear VY:** Verinin lineer olmayan (**hiyerarşik, network**) bir şekilde **saklandığı-erişildiği** **Tree** ve **Graph** gibi yapılardır.



Veri Yapıları Implementasyonu

- VY'leri *implemente* etmek için iki yöntem bulunmaktadır:
 - *Statik veri tipleri* kullanmak (diziler). Statik veri türleri, *kendilerine ayrılan bölümün tamamını* kullanmasalar dahi **sabit** bir **bellek alanı** tahsis ederler.
 - *Dinamik veri tipleri* kullanmak (işaretçi, referans türleri). Program **çalışma zamanında** **boyutları büyüyüp küçülebilir**.





Neler Bilmeliyiz?

- Değişken, Sabit, Operatör Kavramları
- Döngü Denetim Deyimleri
- Fonksiyon Tasarımı, Fonksiyon Çağırma, Parametre Aktarımı
- Diziler ve Bellek Düzeni
- İşaretçiler; İşaretçiler ve Diziler Arasındaki İlişki
- Dinamik Bellek Kullanımı
- Rekürsif Fonksiyonların Çalışması

BİRAZ ARA