



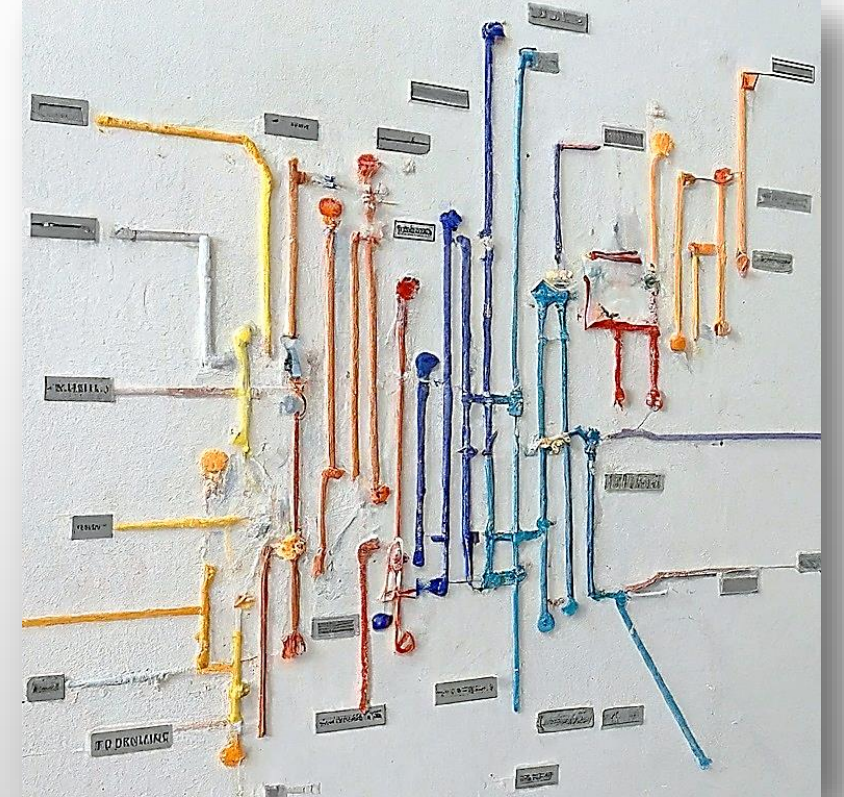
# Bölüm 4: Çizge Algoritmaları

## Algoritmalar



# Çizge Algoritmaları

- Dünya aslında bir ağ gibidir.
  - Şehirler yollarla,
  - İnsanlar ilişkilerle,
  - Bilgisayarlar kablolarla birbirine bağlıdır.
- Çizge algoritmaları bu ağları inceler ve anlamlandırır.





# Çizge Algoritmaları

- Birbirine bağlı noktalar (düğüm) ve bu noktaları birleştiren çizgiler (kenar) ile temsil edilen ağ yapılarını inceler.
- Ağlarda en kısa yolu hesaplama, gruplama gibi işlemleri gerçekleştirir.
- Sosyal ağlar, harita uygulamaları, navigasyon gibi birçok alanda kullanılır.



# Çizge Algoritmalarının Çeşitleri

- Farklı çizge algoritmaları, farklı işlemler için kullanılır.
- Derinlik Öncelikli Arama (DFS):
  - Bir düğümden başlar, dallanarak tüm ağı gezer.
- Genişlik Öncelikli Arama (BFS):
  - Bir düğümden başlar, katman katman tüm ağı gezer.
- Dijkstra Algoritması:
  - Başlangıç düğümünden diğer düğümlere en kısa yolları bulur.
- Kruskal Algoritması:
  - Bir ağı minimum maliyetle birbirine bağlayan kenarları seçer.



# Çizge Algoritmaları

- DFS bir labirentten çıkış yolu ararken kullanılabilir.
- BFS bir haberin tüm şehire yayılma sürecini modelleyebilir.
- Dijkstra en kısa sürede teslimat yapmak için kullanılabilir.





# Çizge Algoritmaları

- Çizge gezinme algoritmaları (*Graph traversal*)
- En kısa yol algoritmaları (*Shortest path*)
- Minimum yayılan ağaç algoritmaları (*Minimum spanning tree*)
- Ağ akış algoritmaları (*Network flow*)





# Çizge Gezinme Algoritmaları (Graph Traversal)

- Çizgenin yapısını sistematik bir şekilde keşfetmek için kullanılır.
- İki ana kategoriye ayrılır:
  - derinlik öncelikli arama (DFS) ve
  - genişlik öncelikli arama (BFS).
- *DFS*, bir düğümden başlayarak mümkün olduğunca derinlere iner ve tüm komşularını ziyaret ettikten sonra geri döner.
- *BFS*, bir kuyruk veri yapısı kullanarak seviye seviye tüm düğümleri ziyaret eder.



# En Kısa Yol Algoritmaları (Shortest Path)

- Başlangıç düğümünden hedef düğüme giden en kısa yolu bulur.
- *Dijkstra*: Başlangıç düğümünden diğer tüm düğümlere olan en kısa yolları bulur. Ağırlıklar pozitif değer olmalıdır.
- *Bellman-Ford*: Negatif ağırlıklı kenar içeren çizgelerde kullanılabilir. Dijkstra Algoritmasından daha yavaştır.
- *A\* Arama*: Sezgisel bilgiler kullanılarak aramayı hızlandırır. Hedef düğüme olan tahmini mesafeyi hesaba katar.





# Minimum Yayılan Ağaç Algoritmaları

- Çizgedeki tüm düğümleri birbirine bağlayan ve toplam kenar ağırlığının en az olduğu alt ağaçtır.
- *Kruskal*, kenarları ağırlıklarına göre sıralar ve döngü oluşturmeyen kenarları seçerek ağacı oluşturur.
- *Prim*, başlangıç düğümünden başlayarak, her adımda en düşük ağırlıklı kenarı seçerek ağacı büyütür.



# Ağ Akış Algoritmaları (Network Flow)

- Ağ üzerinde kaynak noktadan hedef noktaya belirli kısıtlamalar altında akışın nasıl optimize edileceğini modeller.
- Amaç, kısıtlamalar altında kaynak noktadan hedef noktaya mümkün olan en fazla akışı sağlamaktır.
- *Ford-Fulkerson*, en basit ve anlaşılır algoritmalarından biridir, ancak çalışma süresi diğer algoritmalara göre daha uzun olabilir.
- *Edmonds-Karp*, *Ford-Fulkerson* algoritmasını geliştirerek çalışma süresini azaltır.
- *Dinic's*, *Edmonds-Karp* algoritmasından daha hızlı ve bazı durumlarda daha verimlidir.



SON