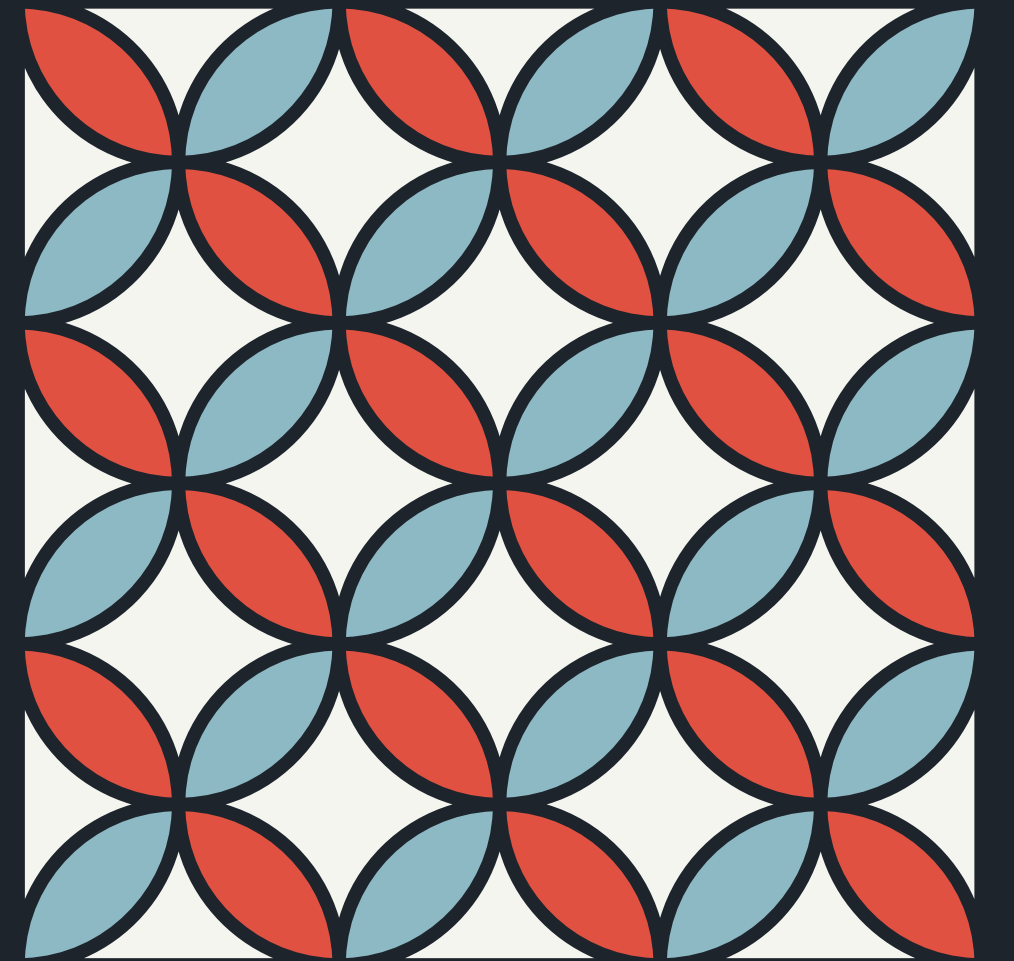
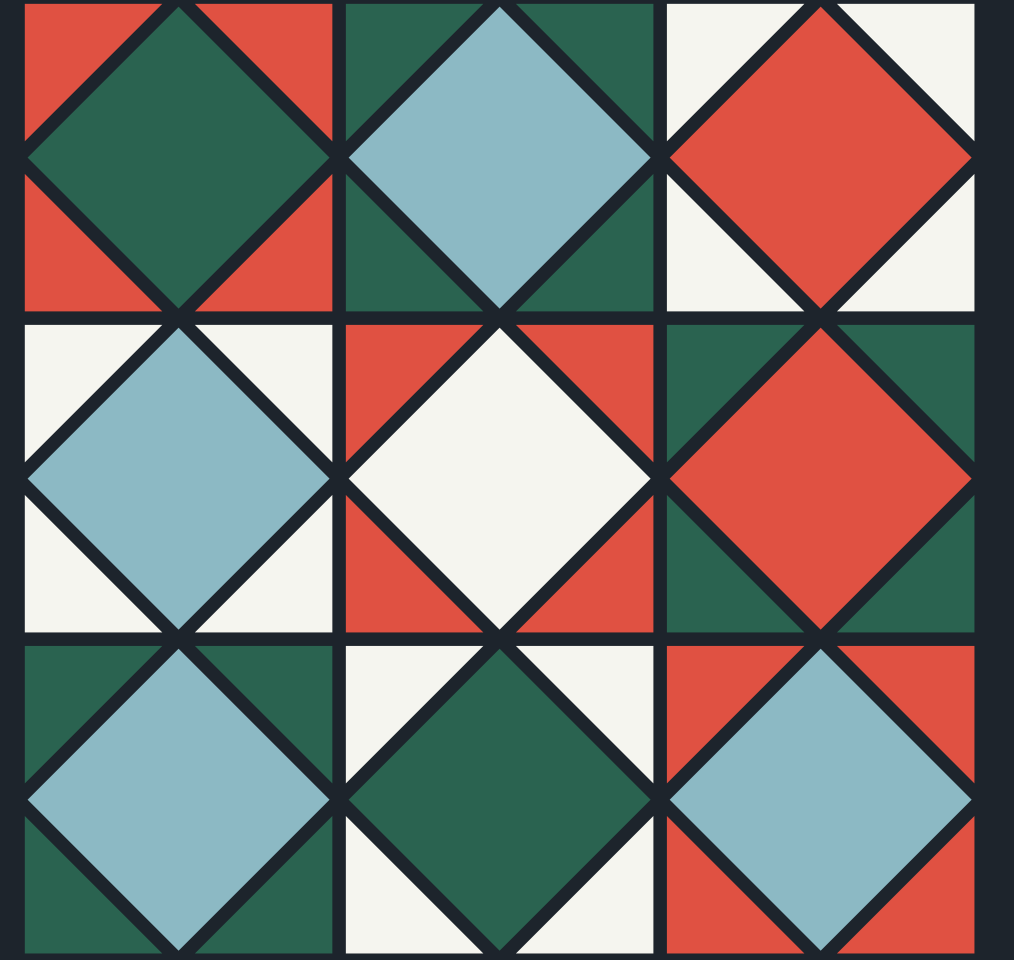
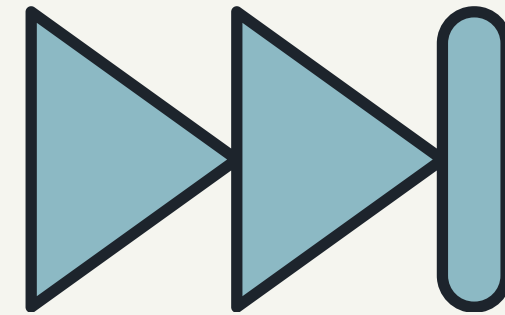


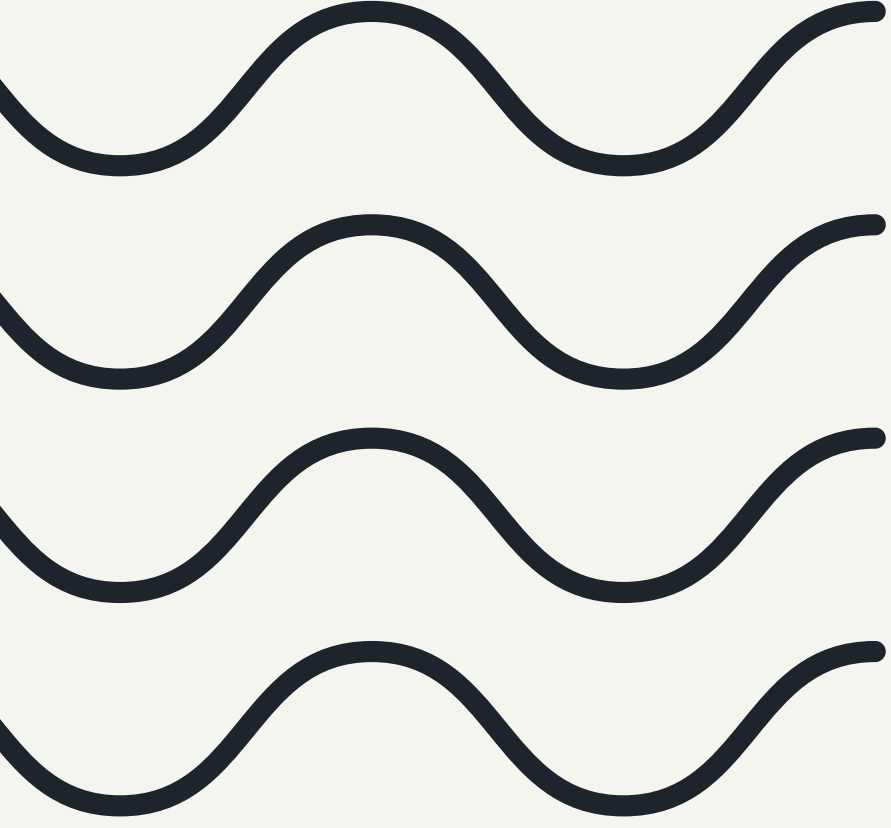
TURKSTUDENTCO

SIRALAMA

ALGORİTMALARI

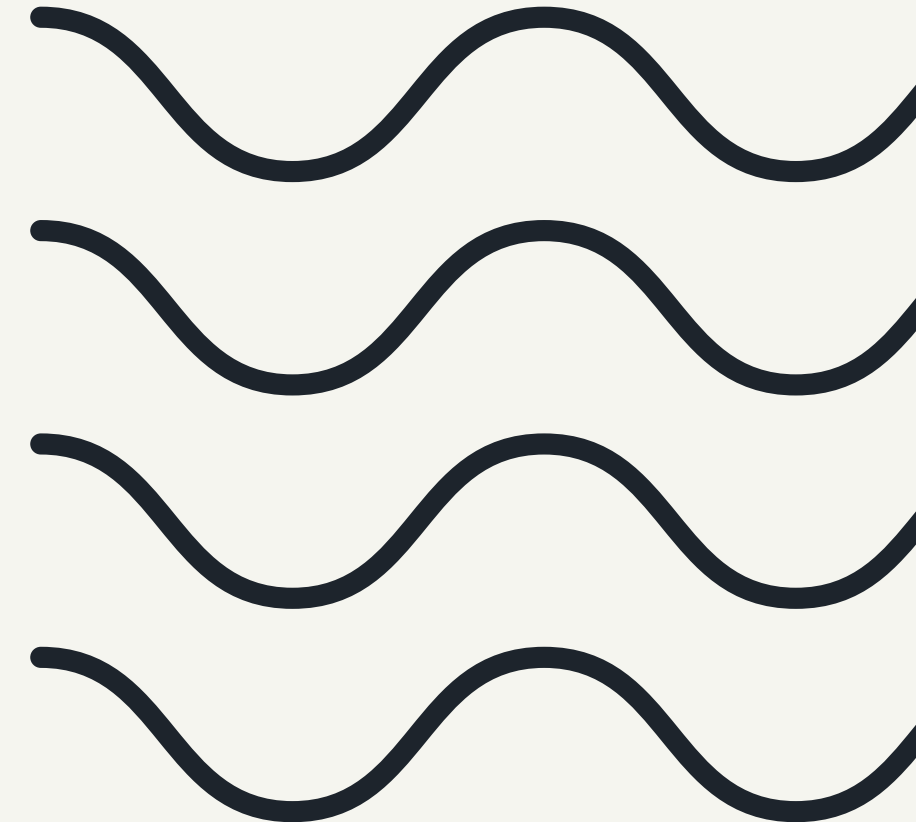
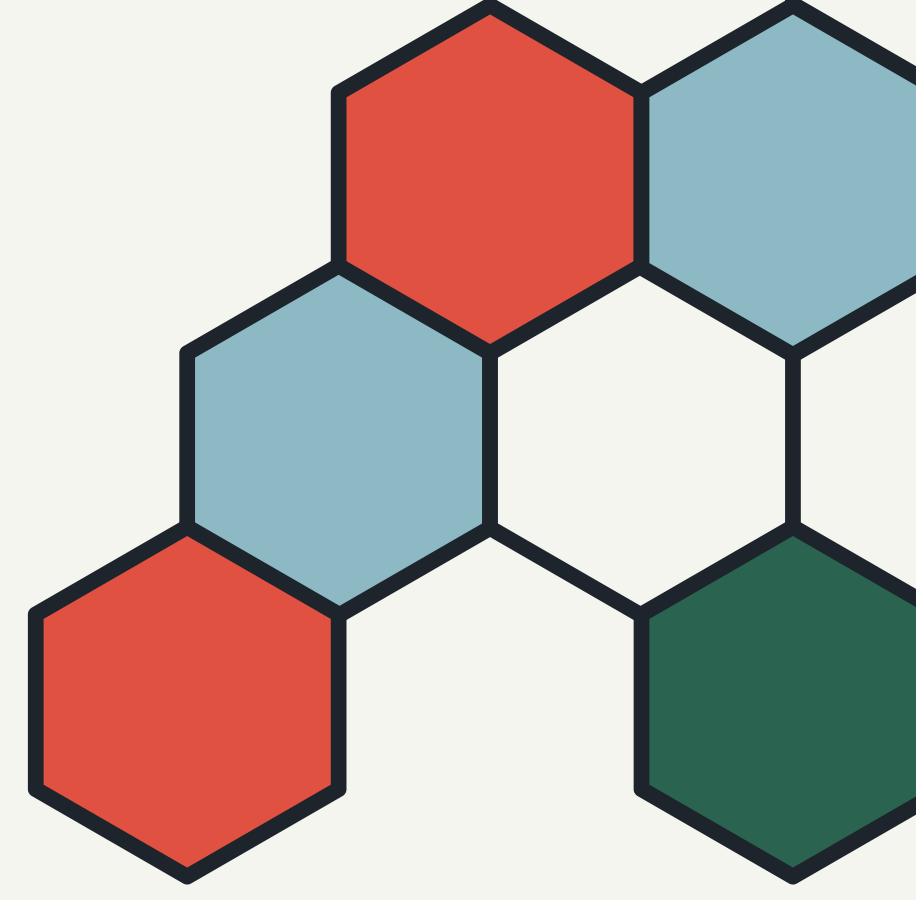
EĞİTMEN
MEHMET ÇELİK





DERS İÇERİKLERİ

- SIRALAMA ALGORİTMALARI
- ARAMA ALGORİTMALARI



SIRALAMA ALGORİTMALARI

Sıralama algoritmaları, bir dizi elemanı belirli bir kritere göre (örneğin sayısal büyüklük, alfabetik sıralama) sıralamak için kullanılan algoritmalarlardır. En yaygın kullanılan sıralama algoritmaları arasında Bubble Sort (Kabarcık Sıralama), Selection Sort (Seçme Sıralama), Insertion Sort (Yerleştirme Sıralama), Merge Sort (Birleştirme Sıralama), Quick Sort (Hızlı Sıralama) e bulunmaktadır. Bu algoritmaların çalışma prensipleri, zaman ve bellek karmaşıklıkları farklıdır.

KABARCİK SIRALAMA(BUBBLE SORT)

- **PRENSİP:** YAN YANA BULUNAN İKİ ELEMAN KARŞILAŞTIRILIR VE YANLIŞ SIRALANMIŞLARSA YERLERİ DEĞİŞTİRİLİR. BU İŞLEM EN BÜYÜK DEĞERİN SONA KADAR "KABARCİK" GİBİ ÇIKMASINA SEBEP OLUR.
- **ZAMAN KARMAŞIKLIĞI:** ORTALAMA VE EN KÖTÜ DURUMDA $O(N^2)$, EN İYİ DURUMDA $O(N)$ (DİZİ ZATEN SIRALIYSA).

SEÇME SIRALAMASI(SELECTION SORT)

PRENSİP: HER ADIMDA DİZİNİN KALAN KISMINDAN EN KÜÇÜK ELEMAN SEÇİLİR VE DİZİNİN BAŞINA YERLEŞTİRİLİR. BU İŞLEM DİZİNİN TAMAMI SIRALANANA KADAR DEVAM EDER.

INSERTION SORT

- **PRENSİP:** DİZİ ÜZERİNDE BİRER BİRER İLERLEYEREK, HER YENİ ELEMAN DAHA ÖNCE SIRALANMIŞ OLAN DİZİ PARÇASINA UYGUN ŞEKİLDE YERLEŞTİRİLİR. BU SÜREÇ, KART OYUNLARINDA KART SIRALAMAYA BENZER.
- **ZAMAN KARMAŞIKLIĞI:** ORTALAMA VE EN KÖTÜ DURUMDA $O(N^2)$, EN İYİ DURUMDA $O(N)$ (DİZİ ZATEN SIRALIYSA).

BİRLEŞTİRME SIRALAMA(MERGE SORT)

- **PRENSİP:** DİVİDE AND CONQUER (BÖL VE FETHET) YAKLAŞIMI KULLANILIR. DİZİ SÜREKLİ İKİYE BÖLÜNÜR VE ARDINDAN BU ALT DİZİLER BİRLEŞTİRİLEREK SIRALANIR.
- **ZAMAN KARMAŞIKLIĞI:** HER DURUMDA $O(N \log N)$, STABİL SIRALAMA ALGORİTMALARINDAN BİRİDİR (EŞİT ELEMANLARIN SIRASI KORUNUR).

HIZLI SIRALAMA(QUICK SORT)

- **PRENSİP:** YİNE DİVİDE AND CONQUER YAKLAŞIMI KULLANILIR. BİR PİVOT ELEMAN SEÇİLİR VE DİZİ, BU PİVOT ELEMANDAN KÜÇÜK VE BÜYÜK OLMAK ÜZERE İKİYE AYRILIR. BU SÜREÇ HER İKİ ALT DİZİ İÇİN DE TEKRARLANIR.
- **ZAMAN KARMAŞIKLIĞI:** ORTALAMA $O(N \log N)$, EN KÖTÜ DURUMDA $O(N^2)$ (KÖTÜ PİVOT SEÇİMİ DURUMUNDA).

ARAMA ALGORİTMALARI



Arama algoritmaları, bir veri yapısında belirli bir öğeyi bulmak veya bir problem için en iyi çözümü keşfetmek amacıyla kullanılan yöntemlerdir. Bu algoritmalar, farklı veri yapıları ve problemlere göre optimize edilir. Arama algoritmaları genel olarak iki kategoriye ayrılabilir: doğrusal (linear) ve dallanma (tree) temelli arama algoritmaları. İşte bazı temel arama algoritmaları:

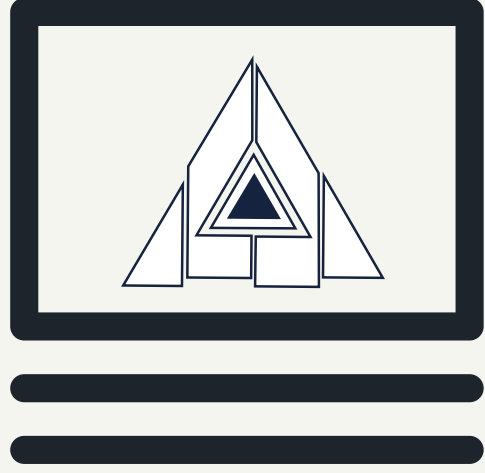
1. Doğrusal Arama (Linear Search)

Tanım: Doğrusal arama algoritması, bir dizideki (veya listede) her bir elemanı sırasıyla kontrol eder ve aranan değeri bulduğunda durur.

Kullanım Durumu: Küçük ve sıralanmamış veri setlerinde etkilidir.

Zaman Kompleksitesi: $O(n)$, burada n eleman sayısıdır.





2. İkili Arama (Binary Search)

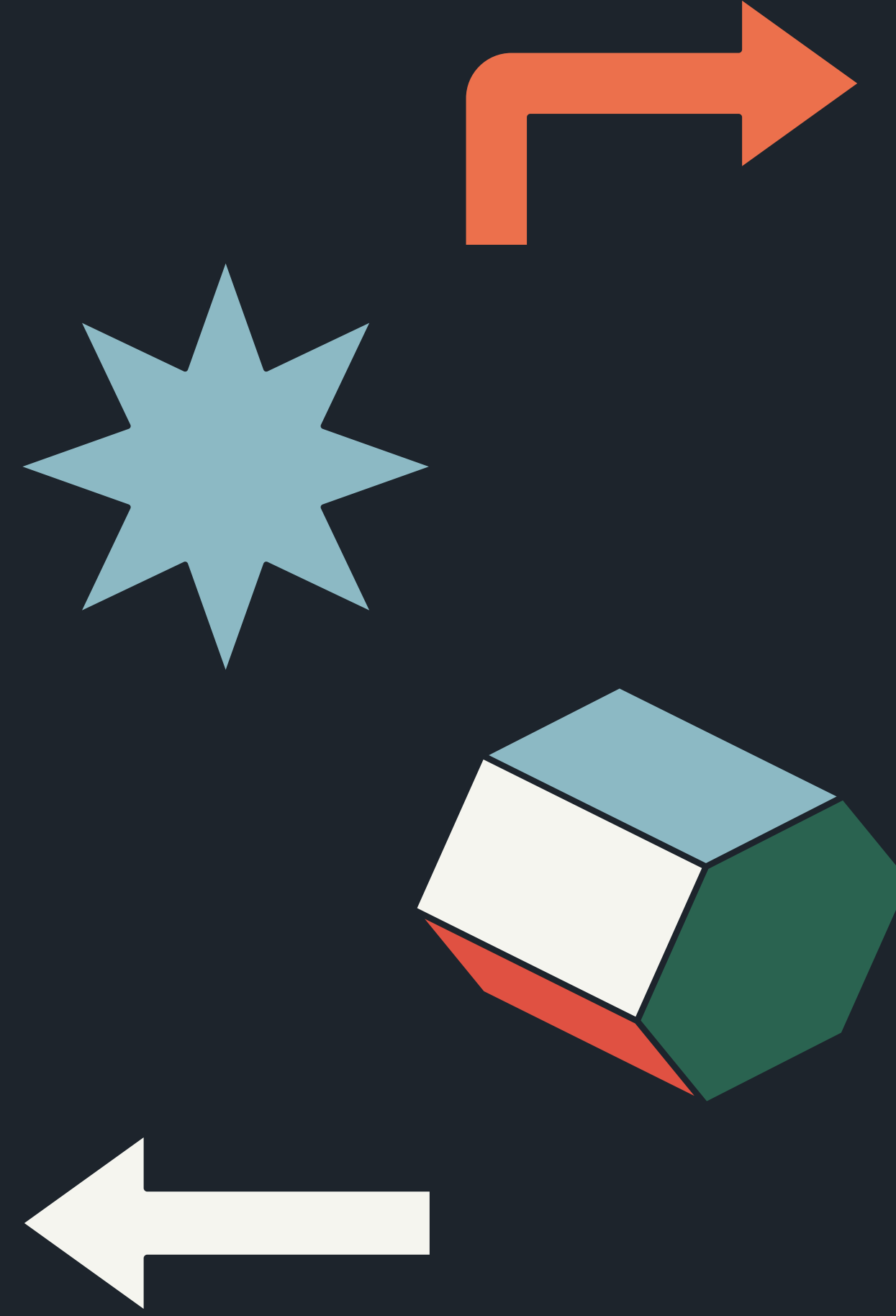
- **Tanım:** İkili arama algoritması, sıralanmış bir dizide çalışır ve aranan öğeyi bulmak için diziyi ikiye böler. Eğer aranan öğe orta noktadan küçükse sol yarıda, büyükse sağ yarıda arama devam eder.
- **Kullanım Durumu:** Sıralı veri setlerinde çok etkilidir.
- **Zaman Kompleksitesi:** $O(\log n)$.
- **Sınırlama:** Dizi sıralı olmalıdır.

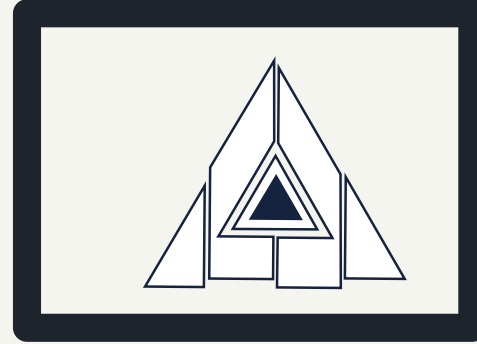
3. Atlamalı Arama (Jump Search)

- **Tanım:** Atlamalı arama algoritması, dizinin belirli aralıklarla (örneğin her \sqrt{n} adımda) kontrol edilerek çalışır ve doğru aralığa ulaşıldığında, o aralıkta doğrusal arama yapılır.
- **Zaman Kompleksitesi:** $O(\sqrt{n})$.
- **Sınırlama:** Dizi sıralı olmalıdır.

4. Dallanma ve Sınırlandırma (Branch and Bound)

- **Tanım:** Bu algoritma, çözüm alanını genişletmek ve sınırlandırmak için kullanılır. Her aday çözüm, belirli sınırlar dahilinde genişletilir veya sınırlandırılır.
- **Kullanım Durumu:** En uygun çözümün aranması gereken durumlarda, özellikle optimizasyon problemlerinde etkilidir.



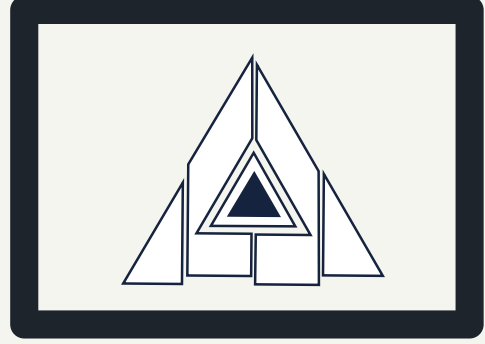


5. A (A Star) Arama*

- **Tanım:** A^* algoritması, hem maliyet (cost) hem de sezgisel bilgi (heuristic) kullanarak hedefe ulaşmak için en kısa yolu arar. En bilinen sezgisel arama algoritmalarından biridir.
- **Kullanım Durumu:** Yol bulma (pathfinding) ve oyunlar gibi durumlarda etkilidir.
- **Zaman Kompleksitesi:** En kötü durumda $O(b^d)$ (burada b dallanma faktörü, d derinliktir).

6. Derinlik Öncelikli Arama (Depth First Search - DFS)

- **Tanım:** DFS, bir graf veya ağacın derinliklerine inerek düğümleri keşfeder ve her bir dalın sonuna ulaşana kadar geri adım atmaz.
- **Kullanım Durumu:** Karmaşık grafiklerde tüm yolların araştırılması gereken durumlarda etkilidir.
- **Zaman Kompleksitesi:** $O(V + E)$ (burada V düğüm sayısı, E kenar sayısıdır).

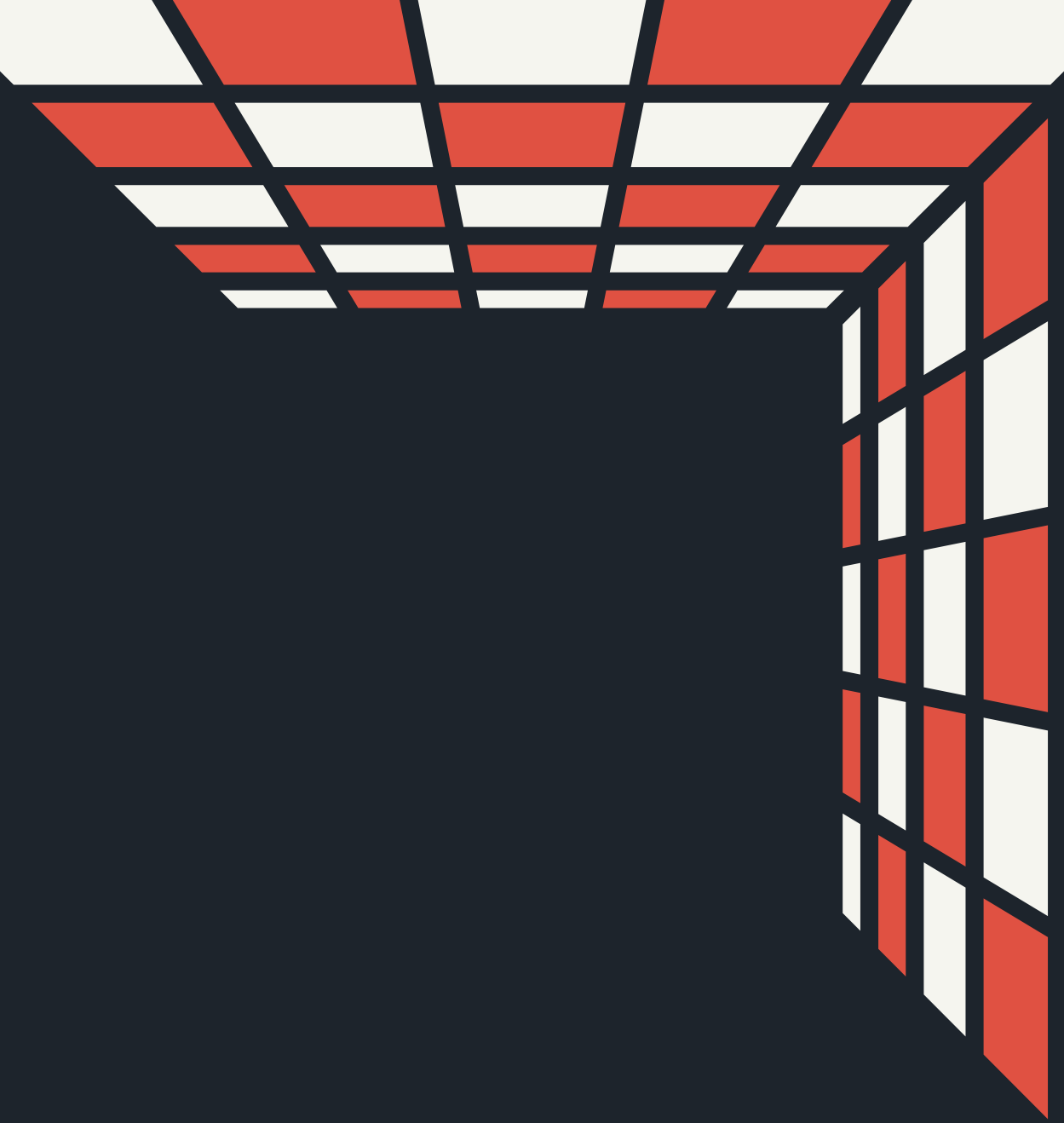


7. Genişlik Öncelikli Arama (Breadth First Search - BFS)

- **Tanım:** BFS, bir graf veya ağacın genişliğine yayılır ve her seviyeyi keşfeder. DFS'nin tersine önce en yakın düğümleri araştırır.
- **Kullanım Durumu:** En kısa yol bulma ve düğümler arasındaki minimum bağlantıyı araştırma problemlerinde kullanılır.
- Zaman Kompleksitesi: $O(V + E)$.

8. İkili Arama Ağacı (Binary Search Tree) Arama

- **Tanım:** İkili arama ağacı, her düğümün sol alt düğümünün kendisinden küçük ve sağ alt düğümünün kendisinden büyük olduğu bir yapıdır. Bu özellikten yararlanılarak, ağacın her bir dalında arama yapılabilir.
- **Zaman Kompleksitesi:** Ortalama $O(\log n)$, ancak dengesiz bir ağaçta $O(n)$ olabilir.





**KATILIMINIZ
İÇİN
TEŞEKKÜRLER**