



# **Bölüm 8: Rastgele Algoritmalar**

## **Algoritmalar**



# Şanslı Algoritmalar

- Bazen problem çözümede şans oyunlarına benzer teknikler kullanılır.
- Karar verme sürecinde rastgelelikten (şans) faydalanılır.





# Rastgele Algoritmalar

- Randomized algorithms
- Algoritma, rasgelelik kullanarak çalışır.
- Çalışma süresi her çalıştırıldığında değişebilir.





# Örnek: Shuffle Algoritması

- Sıralı bir liste elemanlarını karıştırmak için rasgelelik kullanılır.
- Doğruluk:
  - Her elemanın eşit olasılıkla herhangi bir konumda olması garanti edilir.
- Çalışma Zamanı:
  - Her karıştırmada farklı bir süre ortaya çıkabilir.



# Örnek: QuickSort Algoritması

- Pivot seçimi rasgele yapıldığında,
  - algoritmanın ortalama durumu daha iyi olabilir.
- Doğruluk:
  - Sıralı bir dizi için en kötü durumda bile doğru sonuçlar üretir.
- Çalışma Zamanı:
  - Pivot seçimine bağlı olarak değişir.



# BogoSort

- Sıralama algoritmaları arasında en kötü performansa sahip.
- Temel mantığı, liste sıralı olana kadar rasgele bir sıralama yapmaktır.
- Eğlenceli bir algoritma olabilir ancak gerçek dünyada kullanılmaz.
- Listenin elemanlarını rasgele bir şekilde karıştır.
- Liste sıralı mı?
  - Eğer evet, işlemi tamamla.
  - Eğer hayır, adım 1'e geri dön.



# BogoSort'un Zorluğu

- Kombinasyon Sayısı:
  - N elemanlı bir listenin sıralı olma olasılığı  $1/N!$ .
  - Bu nedenle, ortalama durumda çok uzun süreler gerekebilir.
- Kötü Performans:
  - Ortalama ve en kötü durum karmaşıklığı  $O((N+1)!)$ .

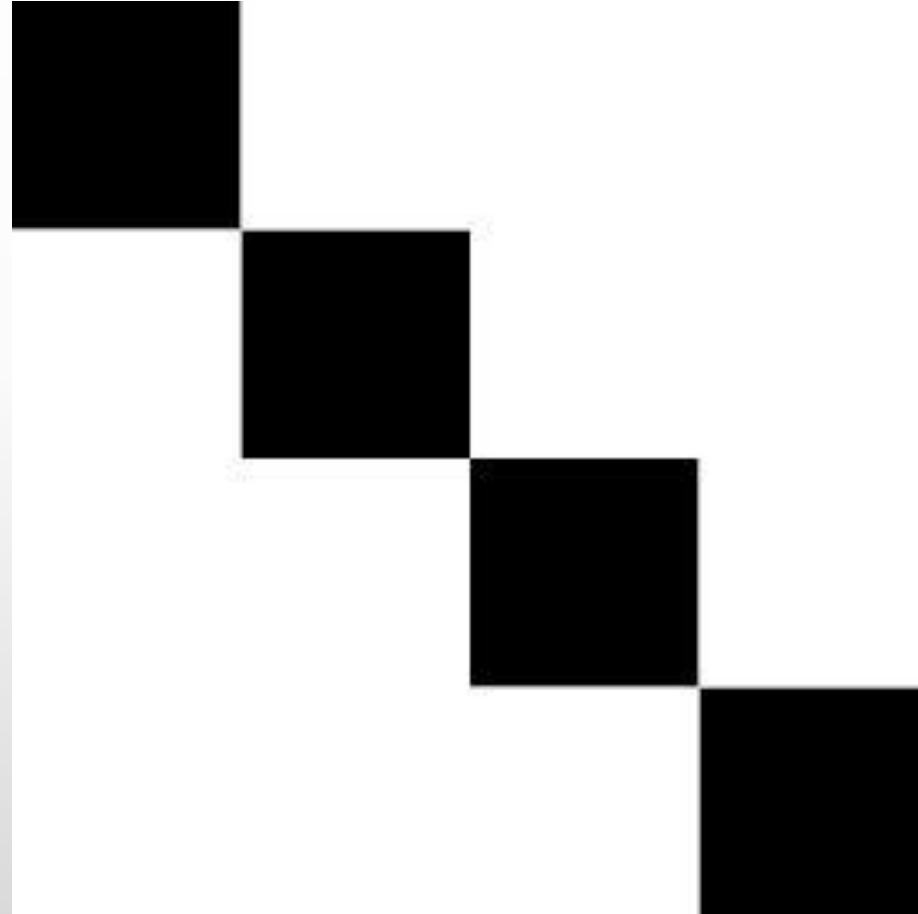


# BogoSort Örneđi

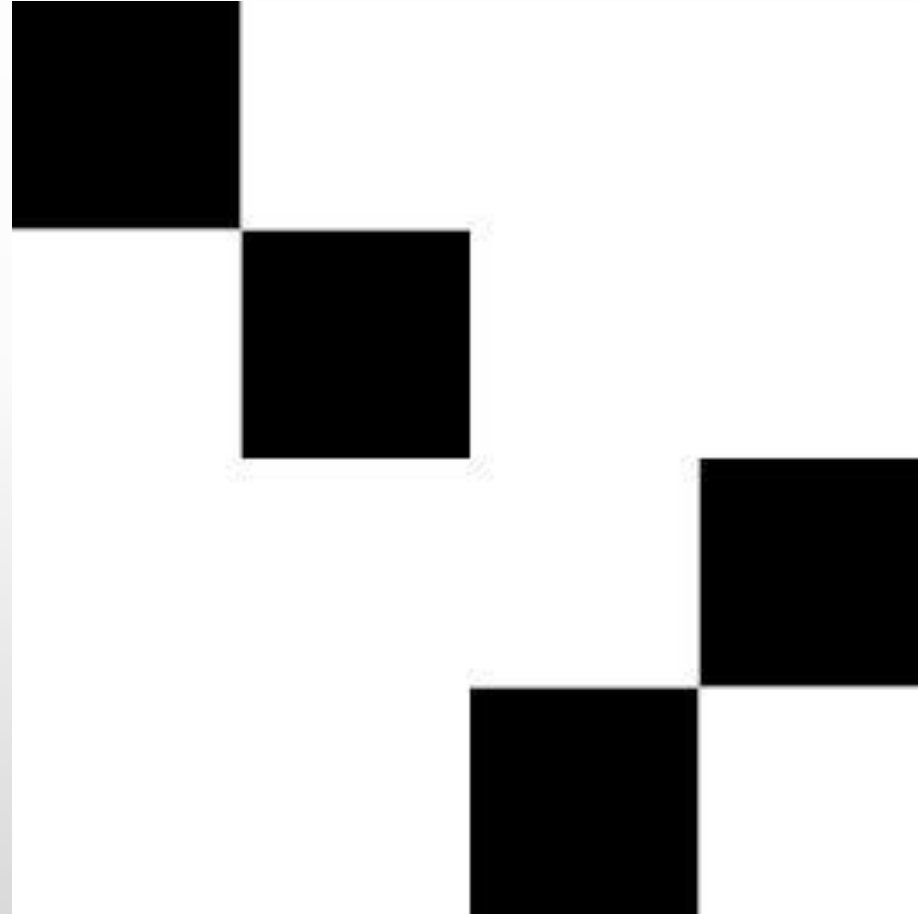
- [5, 2, 8, 4] listesini sırala.
- Adım 1: [4, 8, 2, 5]
- Adım 2: [2, 4, 5, 8]
- Adım 3: [5, 2, 8, 4] (Sıralı deđil)
- Adım 4: [8, 4, 5, 2]
- Adım 5: [2, 5, 8, 4]
- ...



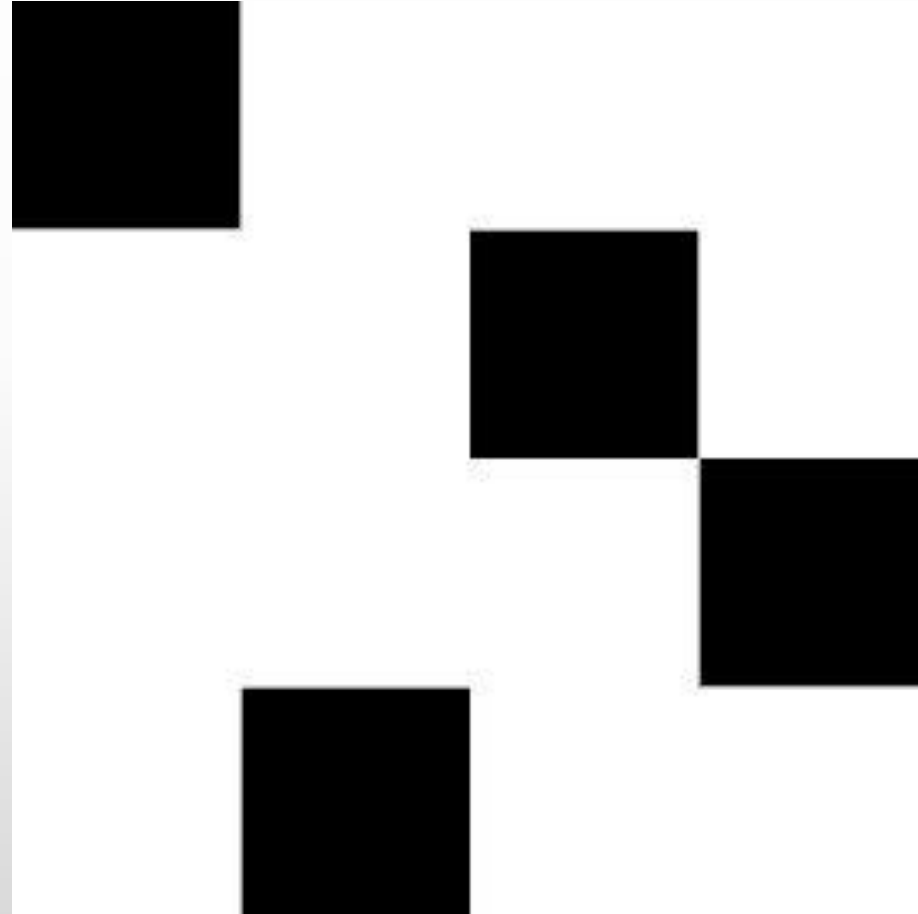
# BogoSort



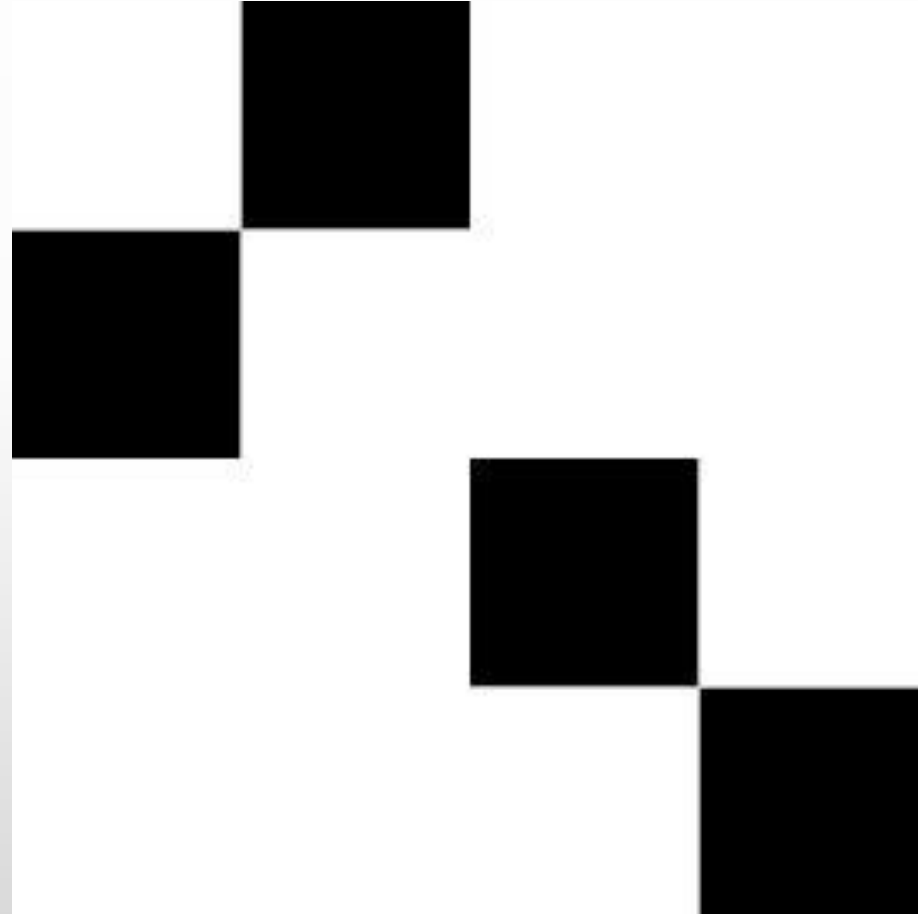
# BogoSort



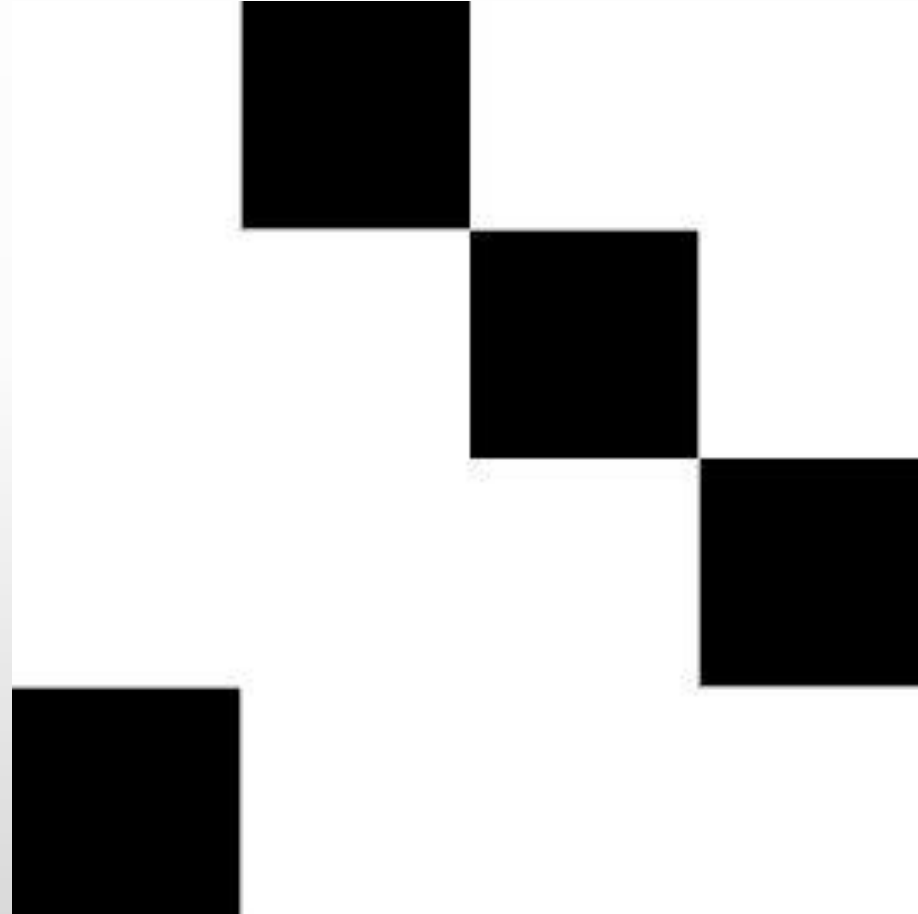
# BogoSort



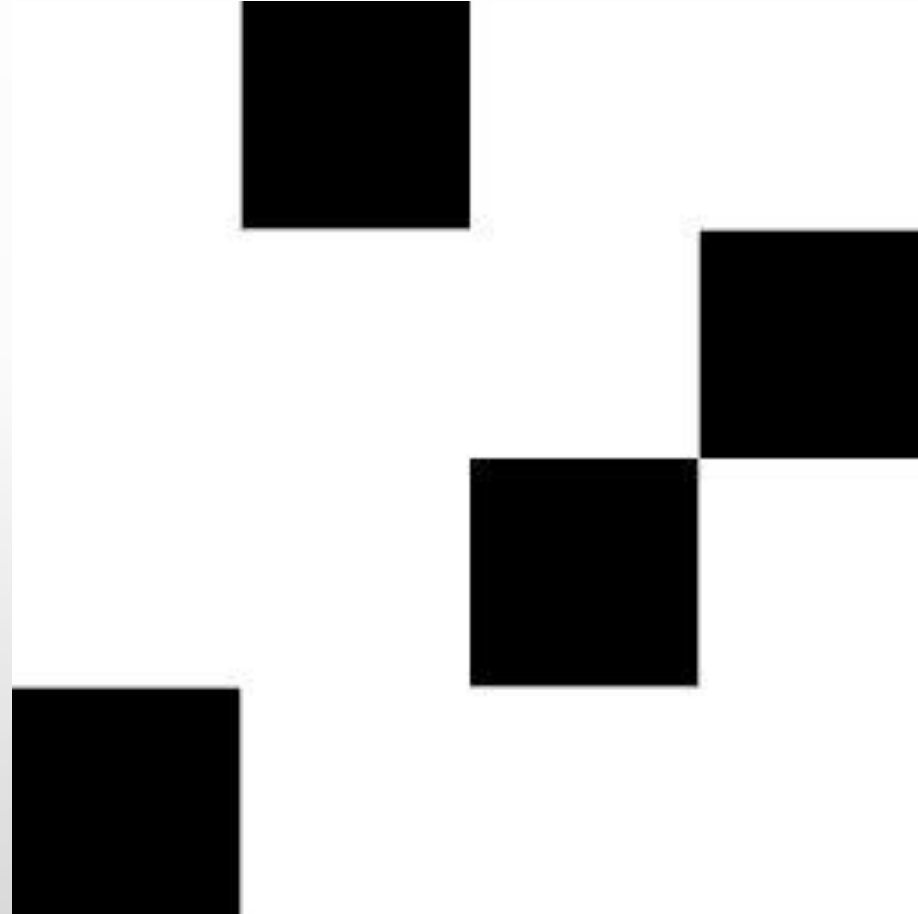
# BogoSort



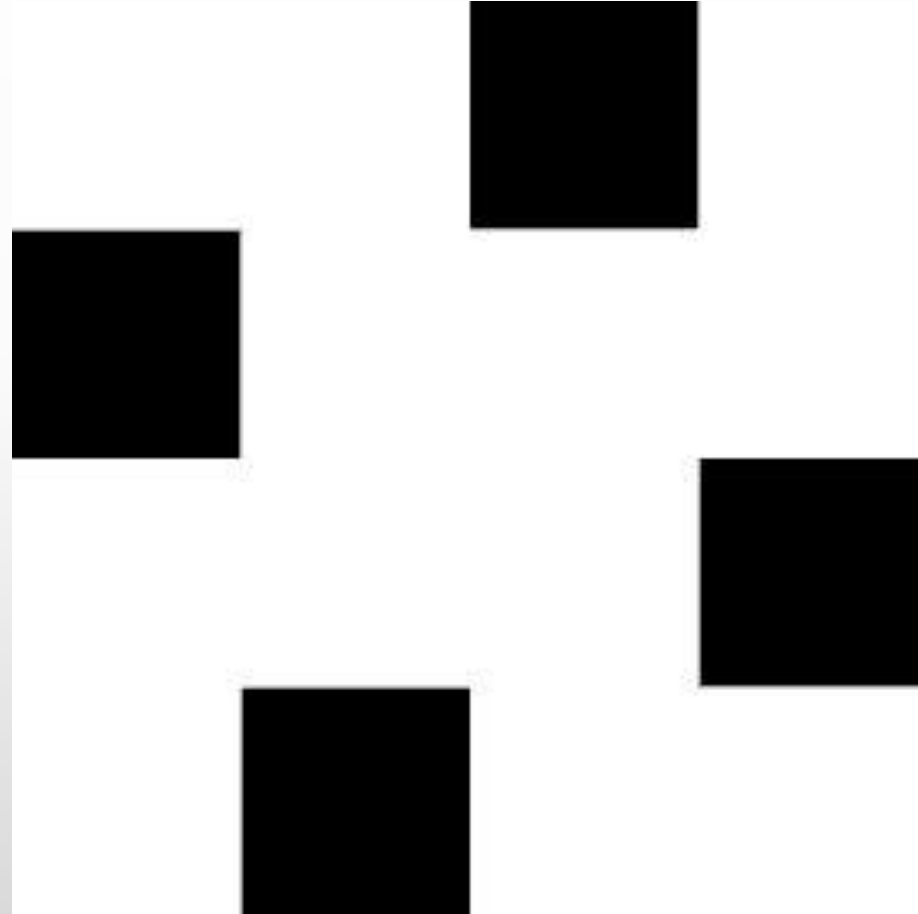
# BogoSort



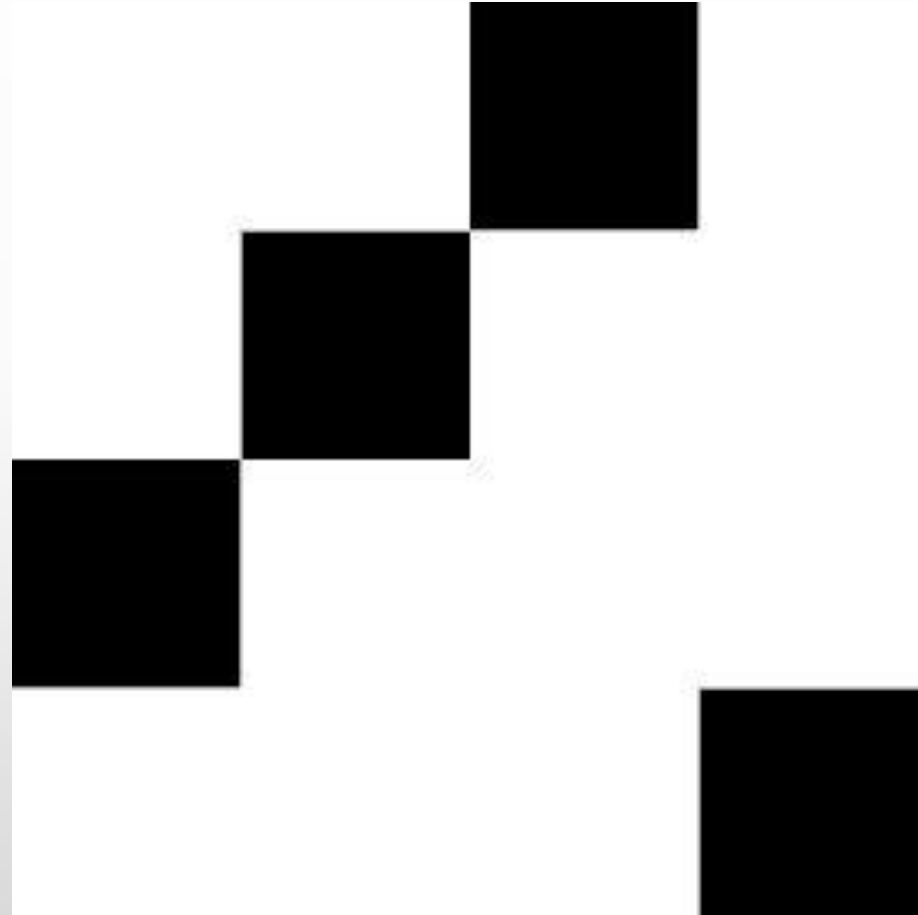
# BogoSort



# BogoSort

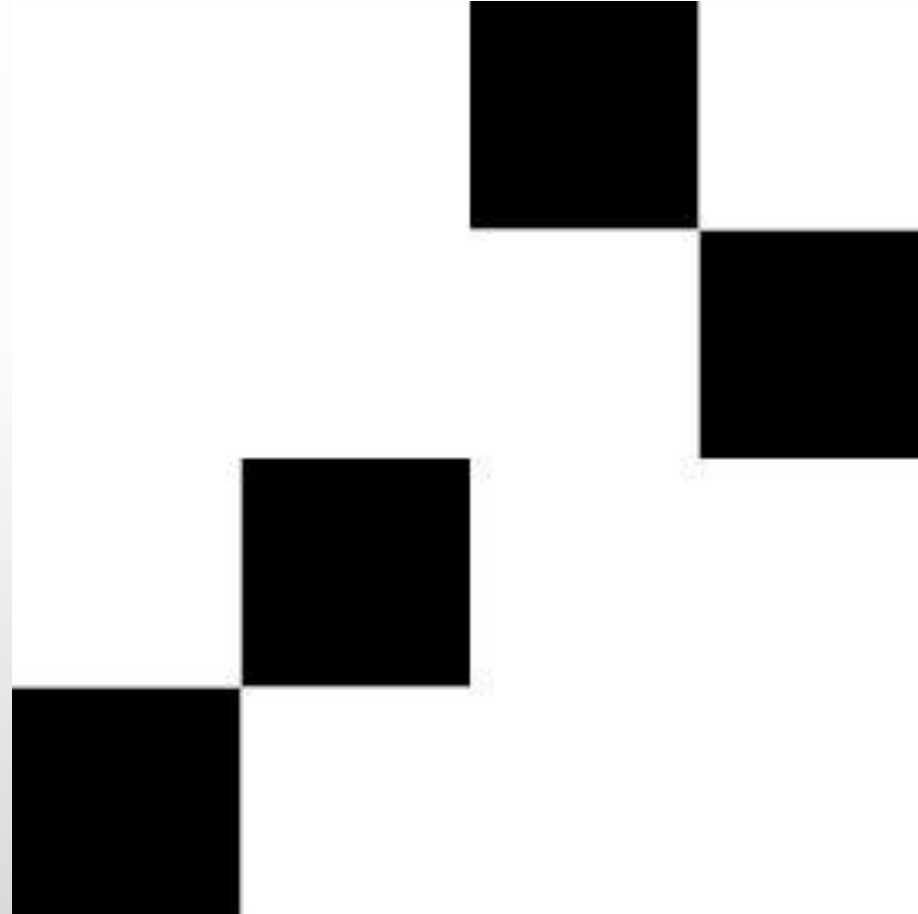


# BogoSort

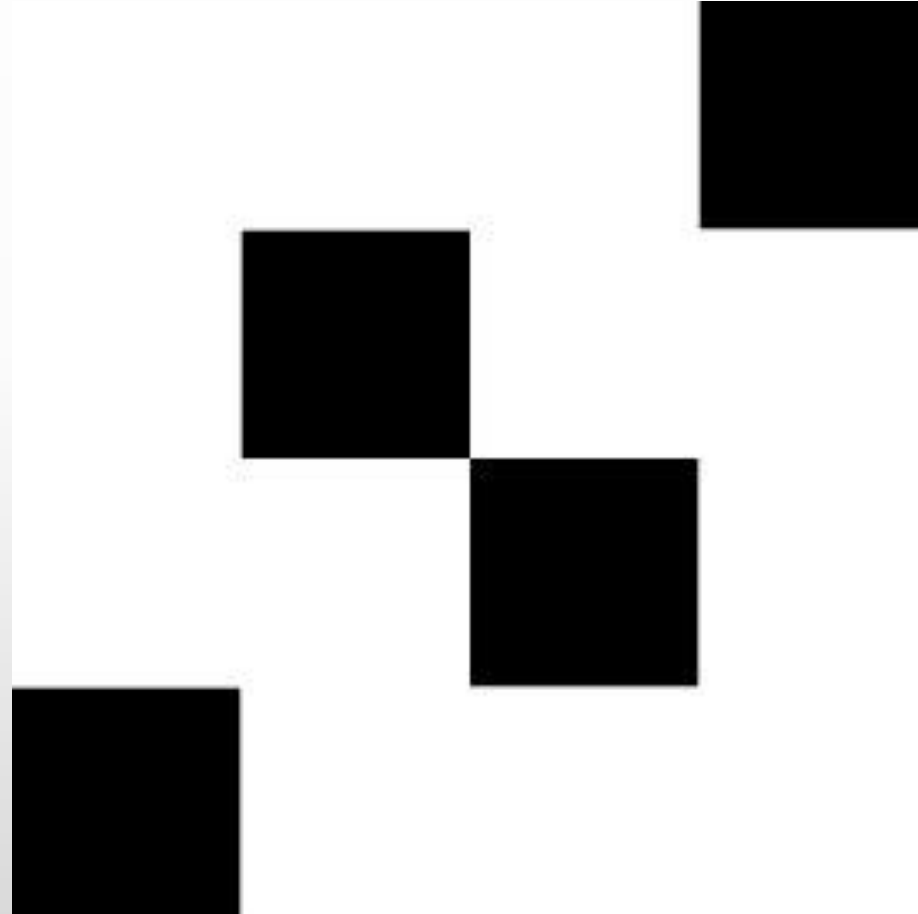




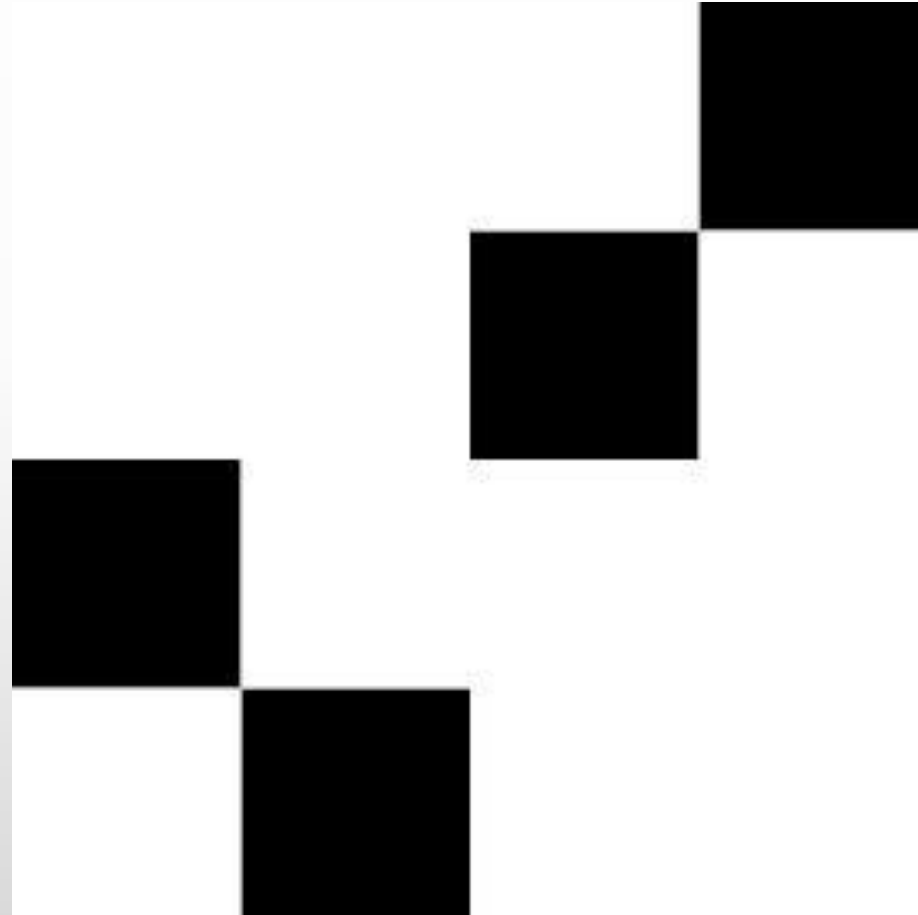
# BogoSort



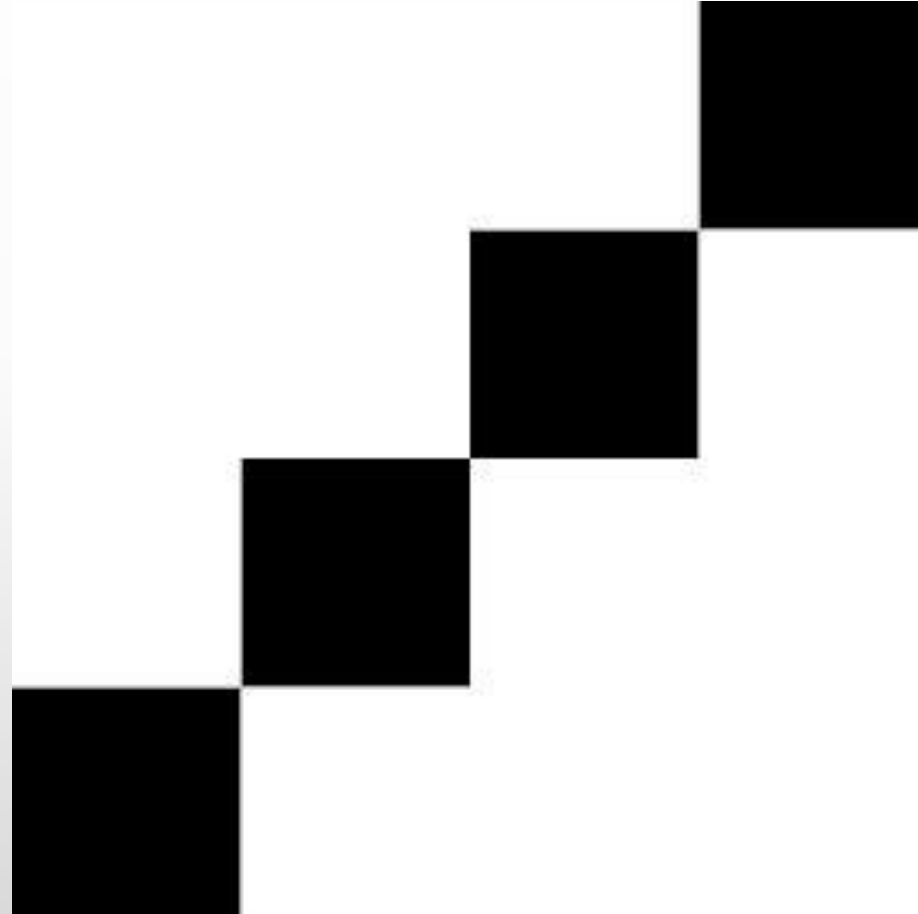
# BogoSort



# BogoSort



# BogoSort







# QuickSort Algoritması

- Parçalama ve bölme işlemleri ile çalışarak hızlı bir şekilde sıralama yapar.
- Algoritmanın Temel Mantığı
  - Bir pivot eleman seçilir.
  - Pivot'tan küçük elemanlar soluna, büyükler sağa yerleştirilir.
  - Her iki parça için tekrar aynı işlemler uygulanır.
  - Sıralanmış parçalar birleştirilir.



# Pivot Seçimi ve Etkisi

- Pivot seçimi algoritmanın performansını etkiler.
- İyi bir pivot seçimi,
  - algoritmanın ortalama ve en kötü durum performansını iyileştirir.
- Ortalama durum karmaşıklığı  $O(n \log n)$ .
- En kötü durum, kötü pivot seçimleri nedeniyle  $O(n^2)$ .
- Ortanca eleman, rastgele seçim gibi stratejiler uygulanır.
- MergeSort'tan daha fazla yer değiştirmeye neden olabilir.
  - ancak genellikle daha hızlı çalışır.
- BubbleSort gibi basit sıralama algoritmalarından daha hızlıdır.



# QuickSort Adım Adım Örneği

- [7, 2, 1, 6, 8, 5, 3, 4] listesini sırala.
- Pivot: 4
  - Sol: [2, 1, 3, 4]
  - Sağ: [7, 6, 8, 5]
- Pivot: 1
  - Sol: [1]
  - Sağ: [2, 3, 4]
- ...



# QuickSort



6 5 3 1 8 7 2 4

# QuickSort



6 5 **3** 1 8 7 2 4

# QuickSort



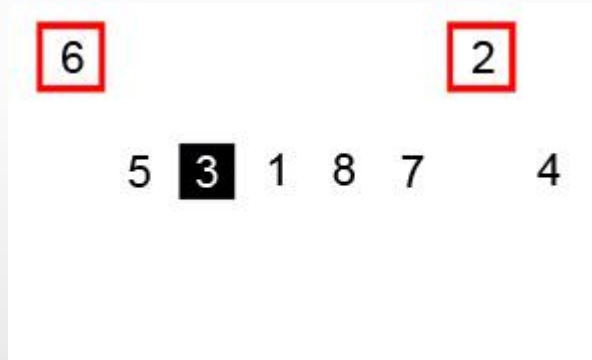
6 5 3 1 8 7 2 4

# QuickSort



6 5 3 1 8 7 2 4

# QuickSort



# QuickSort



2 5 3 1 8 7 6 4

# QuickSort



2 5 3 1 8 7 6 4

# QuickSort



2 5 3 1 8 7 6 4

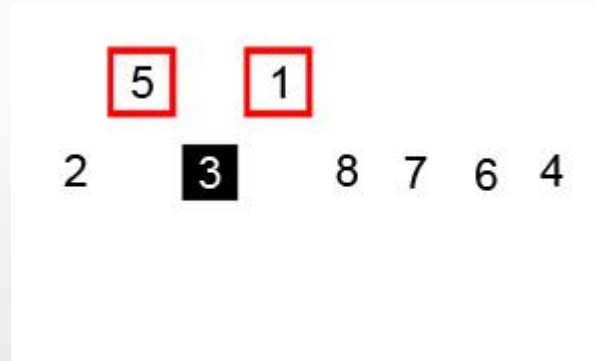


# QuickSort



2 5 3 1 8 7 6 4

# QuickSort



# QuickSort



2 1 3 5 8 7 6 4

# QuickSort



2 1 3 5 8 7 6 4

# QuickSort



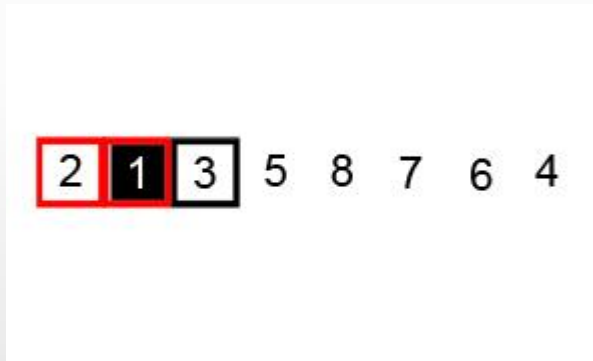
2 1 3 5 8 7 6 4

# QuickSort



2 1 3 5 8 7 6 4

# QuickSort



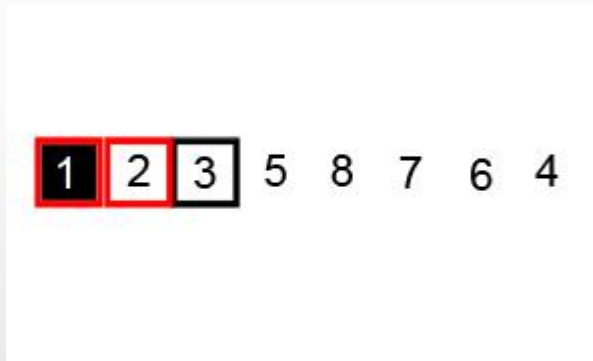
# QuickSort



2 1  
3 5 8 7 6 4



# QuickSort



# QuickSort



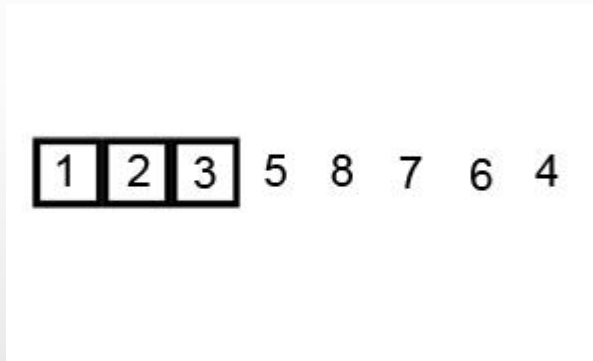
1 2 3 5 8 7 6 4

# QuickSort

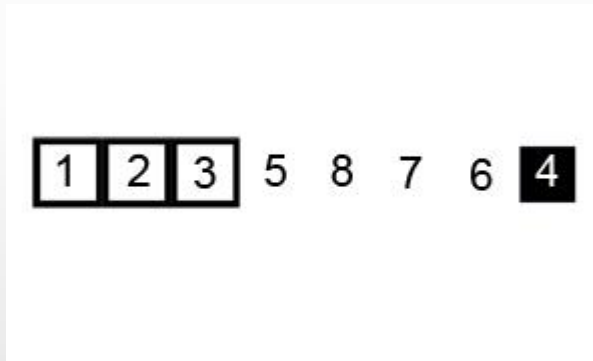


1 2 3 5 8 7 6 4

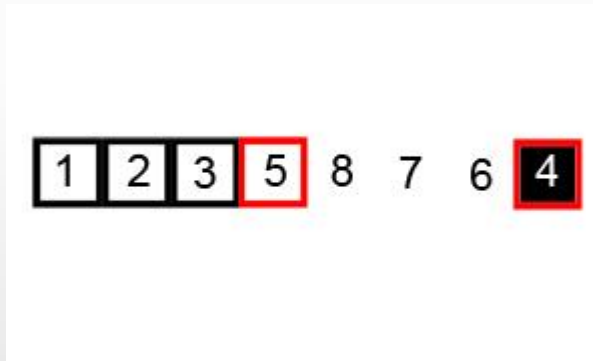
# QuickSort



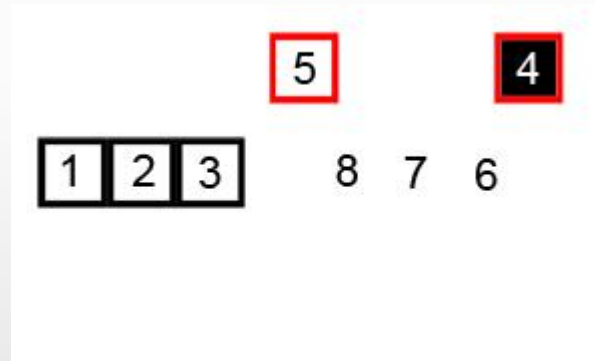
# QuickSort



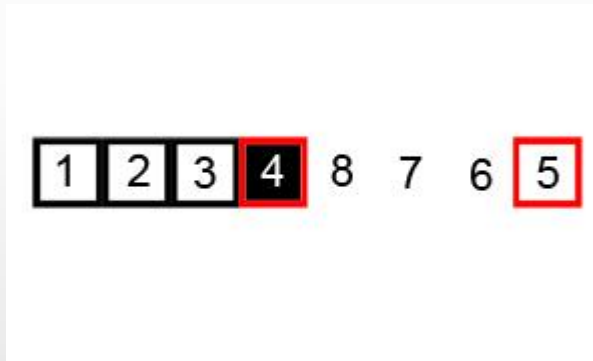
# QuickSort



# QuickSort

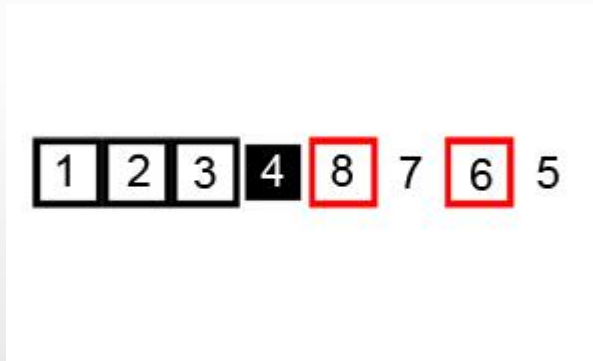


# QuickSort

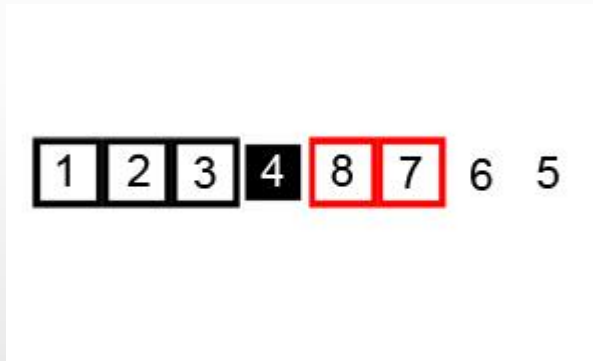




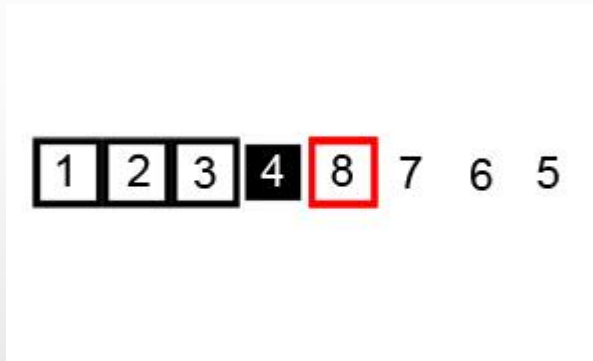
# QuickSort



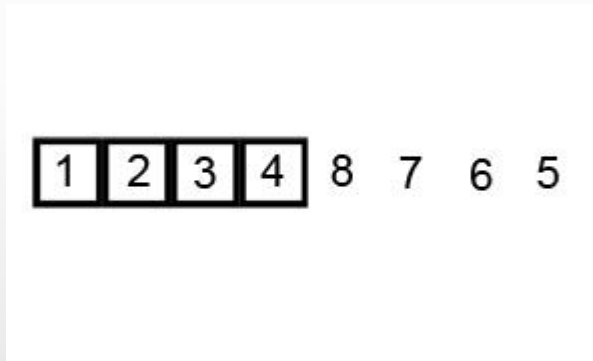
# QuickSort



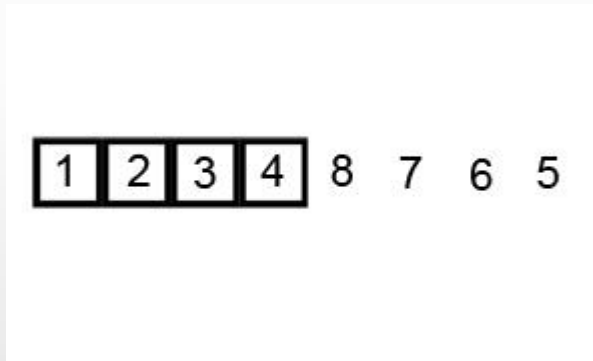
# QuickSort



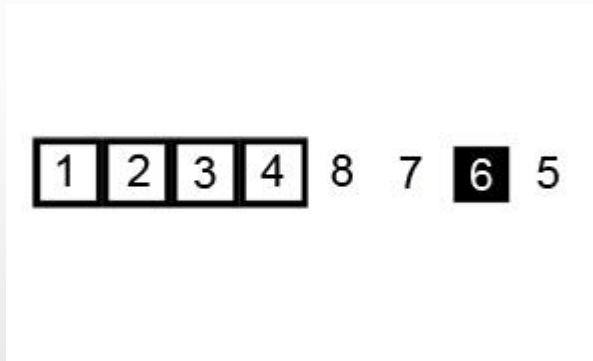
# QuickSort



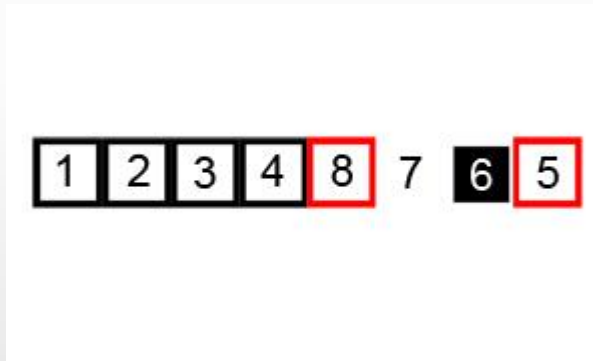
# QuickSort



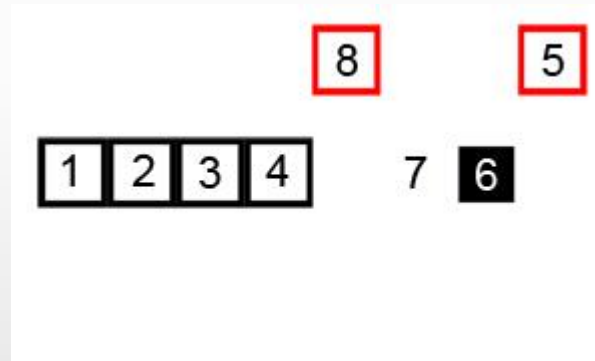
# QuickSort



# QuickSort

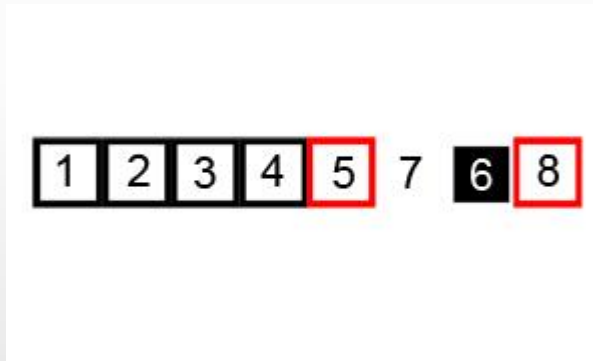


# QuickSort

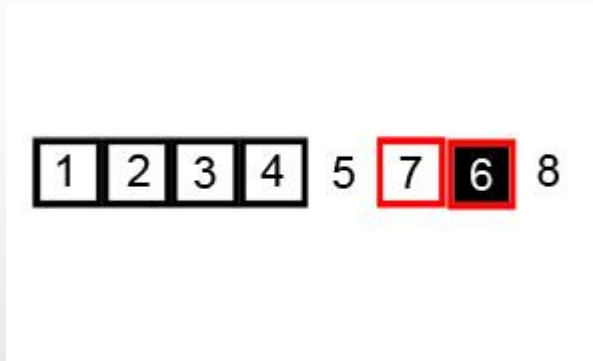




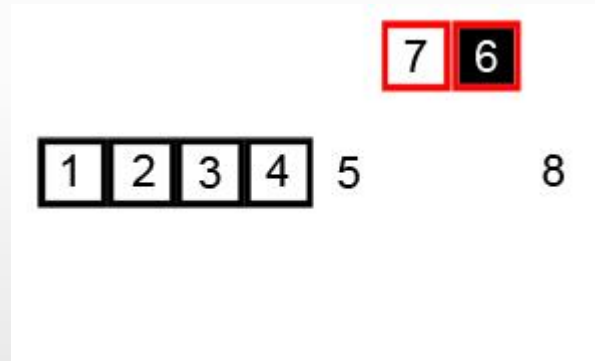
# QuickSort



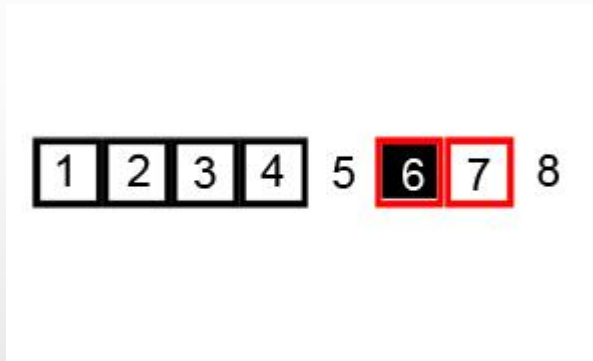
# QuickSort



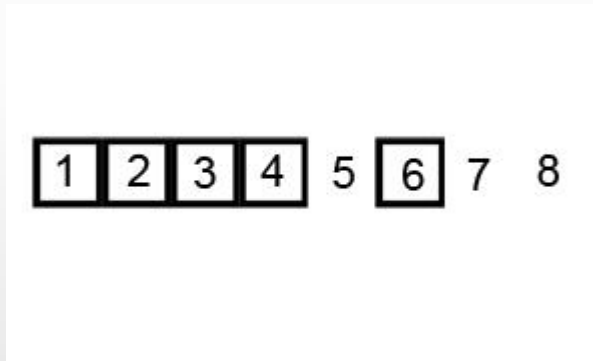
# QuickSort



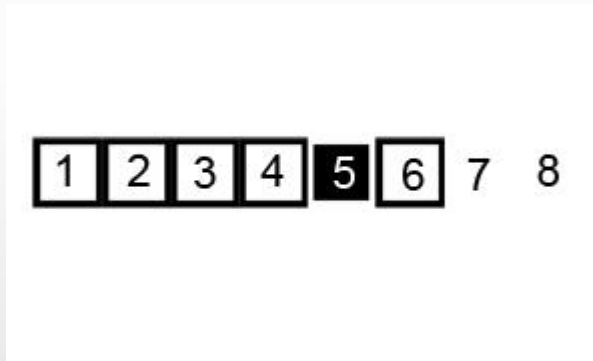
# QuickSort



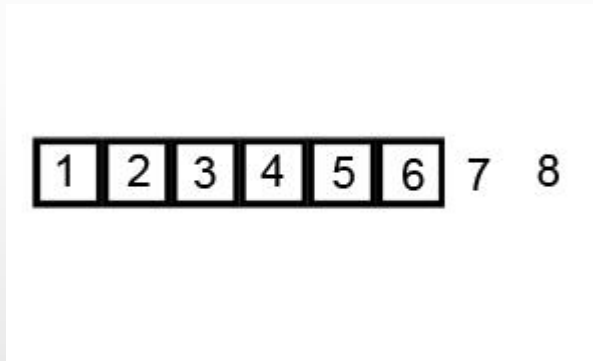
# QuickSort



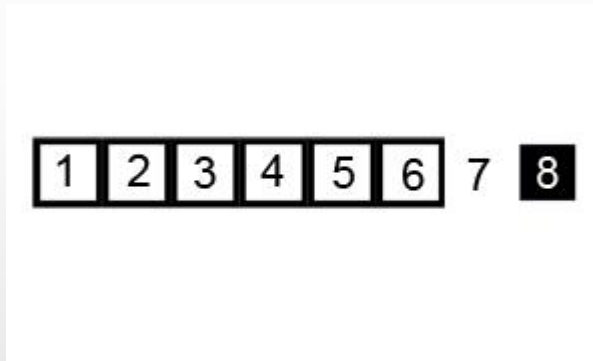
# QuickSort



# QuickSort

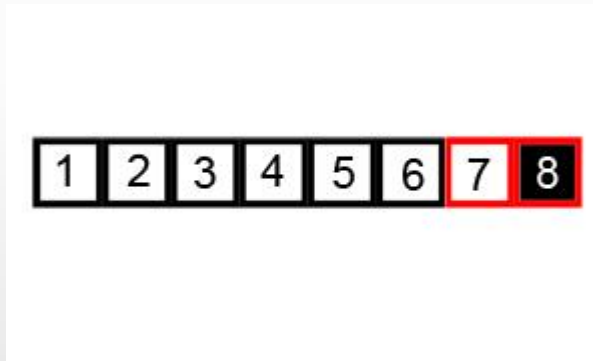


# QuickSort

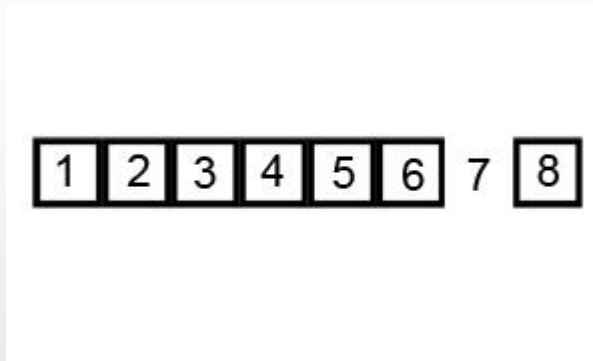




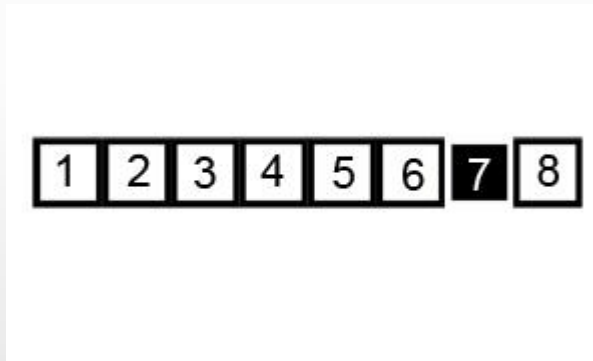
# QuickSort



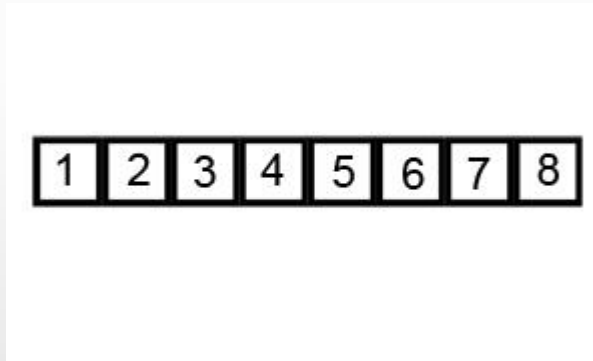
# QuickSort



# QuickSort



# QuickSort







SON