

Ödev 2

**Öğrenci Adı: Emir Kerem Öztürk Öğrenci Numarası: 20011613 Dersin Eğitmeni: M. Elif Karslıgil Video Linki:**

https://drive.google.com/file/d/1GkbC-qlrkKtzSbWO4Wr5Cp1NonXqeBt3/view?usp=sharing

# Problemin Çözümü:

Verilen yönergede, **N**’ye kadar, aynı eleman birden fazla olmayacak şekilde **N** kadar elemandan oluşan bir diziyi **k** sayısında parçaya bölerek sıralama işlemi yapılması isteniliyor. Çözüm için bir **k**- **way** divide and conquer algoritması temel almamız gerekir.

Öncelikle, k değerinin N değerinden büyük olup olunmadığı kontrol edilir. Eğer ki k değeri daha büyük ise, k = N / 2 yapılır bölünme tam olsun diye. Ardından ana dizi **k** adet eşit (**N**/**k**’de kalan var ise eşite yaklaşık) büyüklükte parçalara ayrılır. Bu parçalardan her biri kendi içinde sıralanacaktır. Benim algoritmamda ayrılan parçaların herhangi bir tanesinin eleman sayısı orijinal **k** değerinden küçük olduğu durumlarda, eleman sayısı düşük olan dizileri daha etkili sıralayan insertion sort ile parçalar sıralanmıştır. Daha büyük parçalar rekürsif **k**-**way** merge sort ile insertion sort’un koşuluna uyana kadar küçültülür. Sıralamanın ardından parçaların birleştirilebilmesi için **k**-**way** birleştirme yöntemi kullanılır, yani ilk parçanın min elemanı alınıp diğer her eleman ile karşılaştırılır. En küçük bulunan temp dizisine alınır alınır, min bulunan parçada sonraki indekse geçilir ve yine karşılaştırılır. Merge sort çalıştırılmadan önce zamanlayıcı başlatılır ve dizi sıralanıp birleştirildiğinde durdurulur, böylece algoritmanın hesaplama süresi ölçülür.

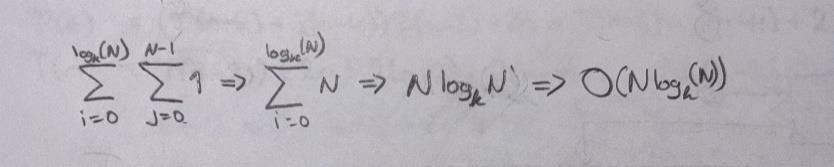
Bu yöntem ile, **k**-way divide and conquer algoritması kullanarak, bize verilen diziyi parçalar halinde sıralayıp verimli bir şekilde tek bir sıralı diziye dönüştürdüm. Insertion sort ve min-heap kullanılarak algoritma optimize edilmiştir. Bu yaklaşım büyük ve rastgele dağıtılmış dizilerde hızlı ve düzenli bir sıralama sağlar. Her **N** için optimal bir **k** değerinin olduğu belli olmuştur.

# Karşılaşılan Sorunlar:

* 1. Algoritmayı yazarken, diziyi **k** parçaya ayırıp parçaları sıraladıktan sonra bu parçaları tek bir dizi haline getirirken zorlukla karşılaştım. Merge sort algoritmaları araştırdım ve min-heap yapısının etkili bir şekilde problemi çözebileceğini gördüm. Her parçanın ilerlemesini aynı zamanda takip edebilmek için indeks ve limit dizilerinin mantığını kavramak zaman aldı.
  2. **N** çok büyük bir sayı verildiğinde, sıralama ve birleştirme işlemleri yüksek bellek ve işlem gücü gerektiriyor. Bellek yetersizliği ile değil ama kodun belli bir **N**’den sonra her **k** için 20-25 saniye üzerinde zaman harcaması ile karşılaştım.

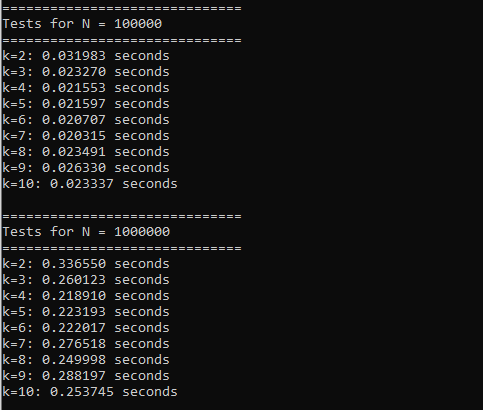
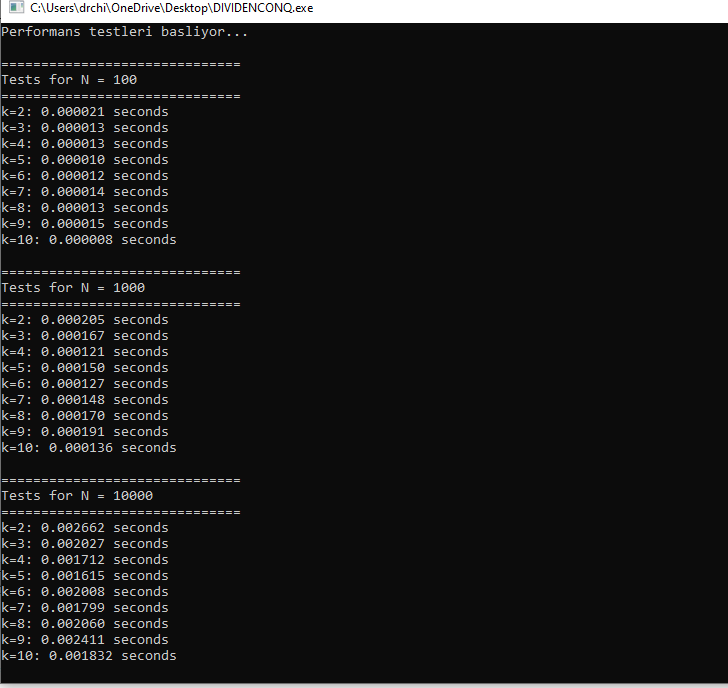
# metin, el yazısı, mektup, harf, yazı tahtası içeren bir resimKarmaşıklık Analizi:

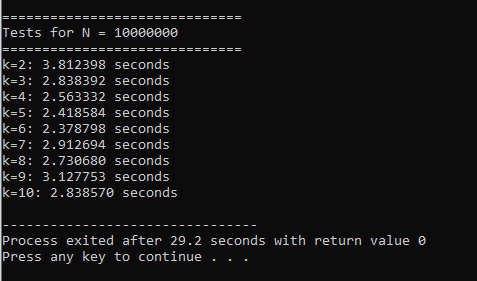
Algoritma da bulunan en çok zaman karmaşıklığına sahip bölge **k**-**way** merge sort fonksiyonudur. Yukarıda pseudo kodu verilen algoritmalar sırasıyla insertion sort, her parçayı k payçaya bölme ve bulunduğu fonksiyonu rekürsif çağırmaktır. Insertion sort **k** üzerindendir.

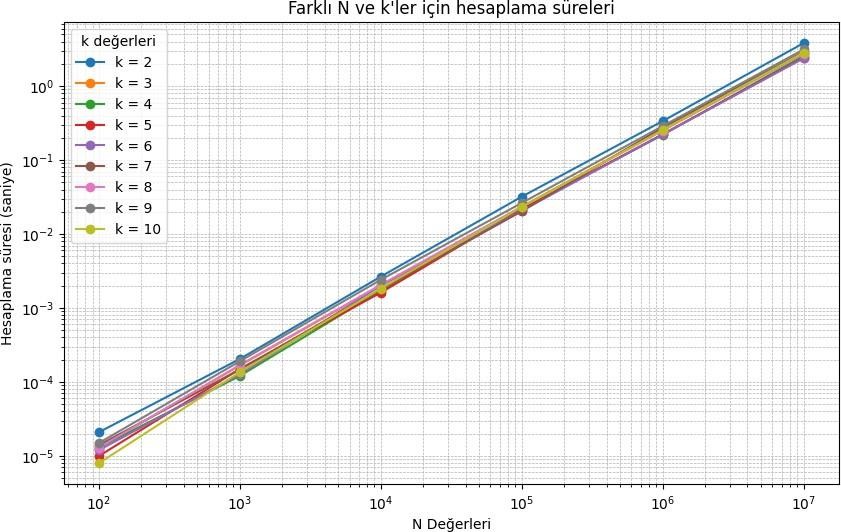


Verilen pseudo kodun karmaşıklığı yukarıdaki denklem ile gösterilmiştir. Bu denklemde iç kısımda kalan bölge açılıp ifade sadeleştirilirse yazdığım algoritmanın karmaşıklığı 𝑂(𝑁 𝑙𝑜𝑔𝑘(𝑁))olur.

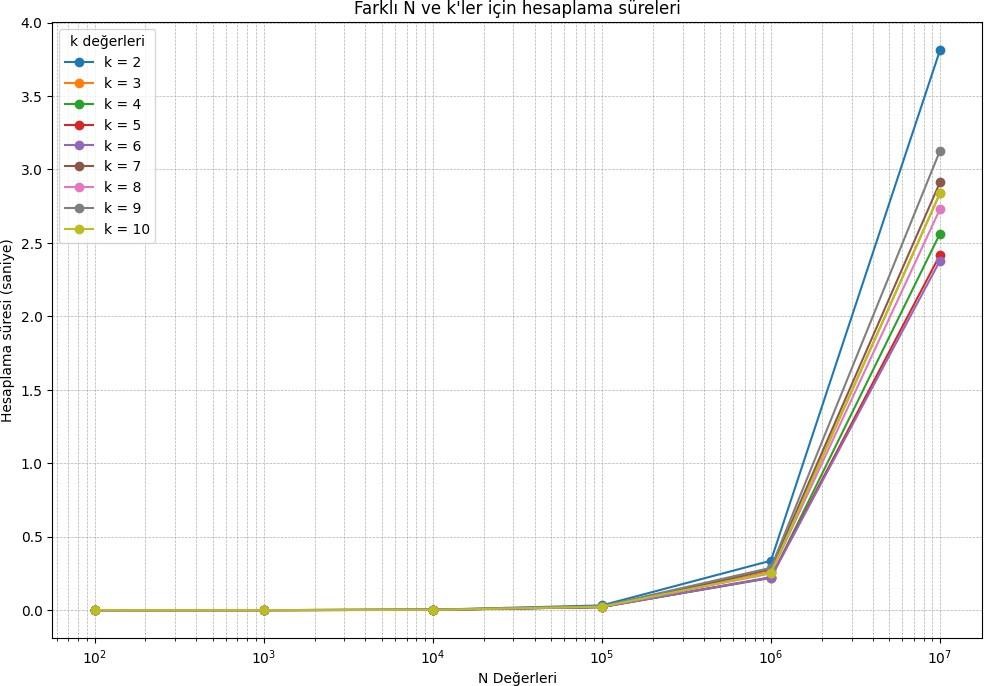
# Ekran Çıktıları:







Yüksek N ve k değerlerine odaklı:



Alçak N ve k değerlerine odaklı:

