

HAZIRLAYAN

**Ceyda Özgenç**

**192802081**

**Ödev1 Konusu**

Radix sort algoritmasını kullanarak verilen dizileri sıralamak ve performans değerlendirmesini yapmak.

**2022/Aralık**

**Radix Sort Algoritması Nedir? Nasıl Çalışır?**

**Radix** sıralama, giriş numaralarını birer birer *basamak* sıralayarak çalışır .

Örnek olarak bu diziyi sıralayalım:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 127 | 324 | 173 | 4 | 38 | 217 | 134 |

İlk önce, birinin yerine göre sıralayacağız:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 127 | 324 | 173 | 4 | 38 | 217 | 134 |

Herhangi bir kararlı sıralama algoritması burada kullanılabilir. Pratikte, sayma sırlaması iyi sonuç verir, çünkü rakamlar sadece az sayıda değer alabilir (örneğin: 10 tabanlı sayılar için 0- 9; 2 tabanlı sayılar için 0- 1).

İşte kişinin yerine göre sıralama bize ne getiriyor:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 127 | 324 | 173 | 4 | 38 | 217 | 134 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 173 | 324 | 4 | 134 | 127 | 217 | 38 |

Ardından, onluk yerine göre sıralayacağız:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 173 | 324 | 04 | 134 | 127 | 217 | 38 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 04 | 217 | 324 | 127 | 134 | 38 | 173 |

Onlar hanesinde bir eşitlik olduğunda, birler basamağı daha düşük olan sayının önce geldiğine dikkat edin. Bu nedenle kararlı bir sıralama algoritması kullanmak önemlidir. Bu, mevcut basamakta bir bağ varsa, zaten sıraladığımız basamaklara göre sayıların nasıl sıralandığına geri döneceğimiz anlamına gelir.

Ve son olarak, yüzler yerine göre sıralayacağız:

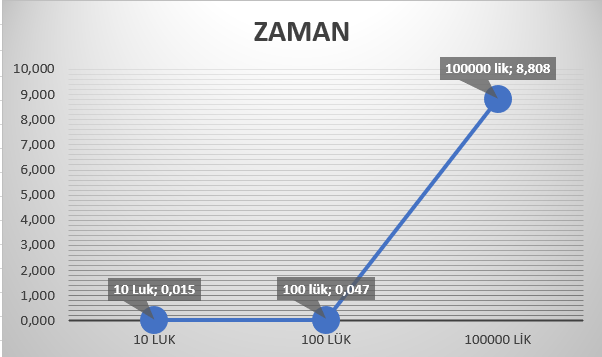
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 004 | 217 | 324 | 127 | 134 | 38 | 173 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 004 | 038 | 127 | 134 | 173 | 217 | 324 |

Sonuç olarak;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 004 | 038 | 127 | 134 | 173 | 217 | 324 |

**Boyut-Zaman Grafikleri**



**Boyut-zaman Grafiklerinin Yorumlanması**

**Radix algoritmasının en iyi ve en kötü zaman karmaşıklıkları nedir?**

## En kötü durum zaman karmaşıklığı

Sayı tabanı sıralamasındaki en kötü durum, önemli ölçüde fazla sayıda basamağa sahip bir öğe dışında tüm öğeler aynı sayıda basamağa sahip olduğunda ortaya çıkar. En büyük öğedeki basamak sayısı n'ye eşitse, çalışma zamanı O(n 2 ) olur. Sayma sıralamasının en kötü çalışma süresi O(n+b)'dir. b=O(n) ise, en kötü durum çalışma süresi O(n)'dir. Burada, countingSort işlevi d kez çağrılır, burada d =⌊benÖGB(mx)+1⌋⌊benÖGB(mx)+1⌋.

Sayı tabanı sıralamasının toplam en kötü durum karmaşıklığı O(log b (mx)(n+b)) şeklindedir.

## En iyi durum zaman karmaşıklığı

En iyi durum, tüm öğelerin aynı sayıda basamağa sahip olması durumunda ortaya çıkar. En iyi durum zaman karmaşıklığı O(d(n+b))'dir. b = O(n) ise, zaman karmaşıklığı O(dn) olur.

**Radix Algoritmasının en iyi ve en kötü zaman çalışma zamanı hangi durumlarda oluşur?**

Girilen tamsayılarda d basamak olsun. Radix Sıralaması O(d\*(n+b)) zaman alır, burada b sayıları temsil etmek için tabandır, örneğin ondalık sistem için b, 10'dur. d'nin değeri nedir? Eğer k olası maksimum değer ise, o zaman d O olur(logb (k)). Yani toplam zaman karmaşıklığı O((n+b) \* logb (k))'dir. Bu, büyük bir k için karşılaştırmaya dayalı sıralama algoritmalarının zaman karmaşıklığından daha fazlasını gösterir. Önce k'yi sınırlayalım. k <= nc olsun , burada c bir sabittir. Bu durumda karmaşıklık O(nLogb (n)) olur.

Yani d nin ve k’nın en küçük olduğu durumlar en iyi en büyük olduğu durunlar en kötüsüdür.