

Deterministik sıralama algoritması

23:49 29.12.2024 Ahmet efe

Bu algoritmaya geçmeden önce "DETERMİNİSTİK SİSTEM" kavramının teknik anlamını tanıyalım;

Deterministik sistem matematik ve fizikte, deterministik bir sistem, sistemin gelecekteki durumlarının gelişmesinde rastgelelik bulunmayan bir sistemdir.[1]

Belirli bir model bu nedenle her zaman belirli bir başlangıç koşulundan veya başlangıç durumundan aynı çıktı üretir

Evet şimdi biraz daha açık olursak bu kavram matematiksel / sayısal dünyada kısaca bu şekilde ifade edilebilir Şimdi ise yazılımdaki yerine gelelim;

Yazılımdaki yeri aslında bir çok yer olsada burda bir algoritmayı ele alıcaz. Bu algoritma deterministik kavramına uyan türden bir algoritmadır kısaca;

Belirli bir dizide aynı çıktıyı veren ancak farklı dizilerde farklı çıktıyı veren diziler üzerinde işlemler yapan bir algoritma "Deterministik" kategorisine girer

bu bilgileri göz önünde bulundurarak algoritmaya bir göz atalım;

```

nQ = 256

input_array = [n for n in range(nQ)]
output_array = []

print('Input array: {}'.format(input_array))

if input_array:
    if input_array != []:
        if isinstance(input_array, list):
            for argument in input_array:
                q = [list(str(argument).encode())]
                for deep_argument in q:
                    qout = [x for x in list((str(deep_argument).encode()))]
                    output_array.append([sum(qout) / len(qout), input_array.index(argument)])

new_arr = [input_array[ix[1]] for ix in sorted(output_array)]

print(new_arr)

```

Algoritmada 4 temel ve 2 değişken unsur bulunuyor

TEMEL UNSURLAR

1. *Input array*
2. *Output array*
3. *New array*
4. *nQ*

Değişken Unsurlar

5. *Qout*
6. *Q*

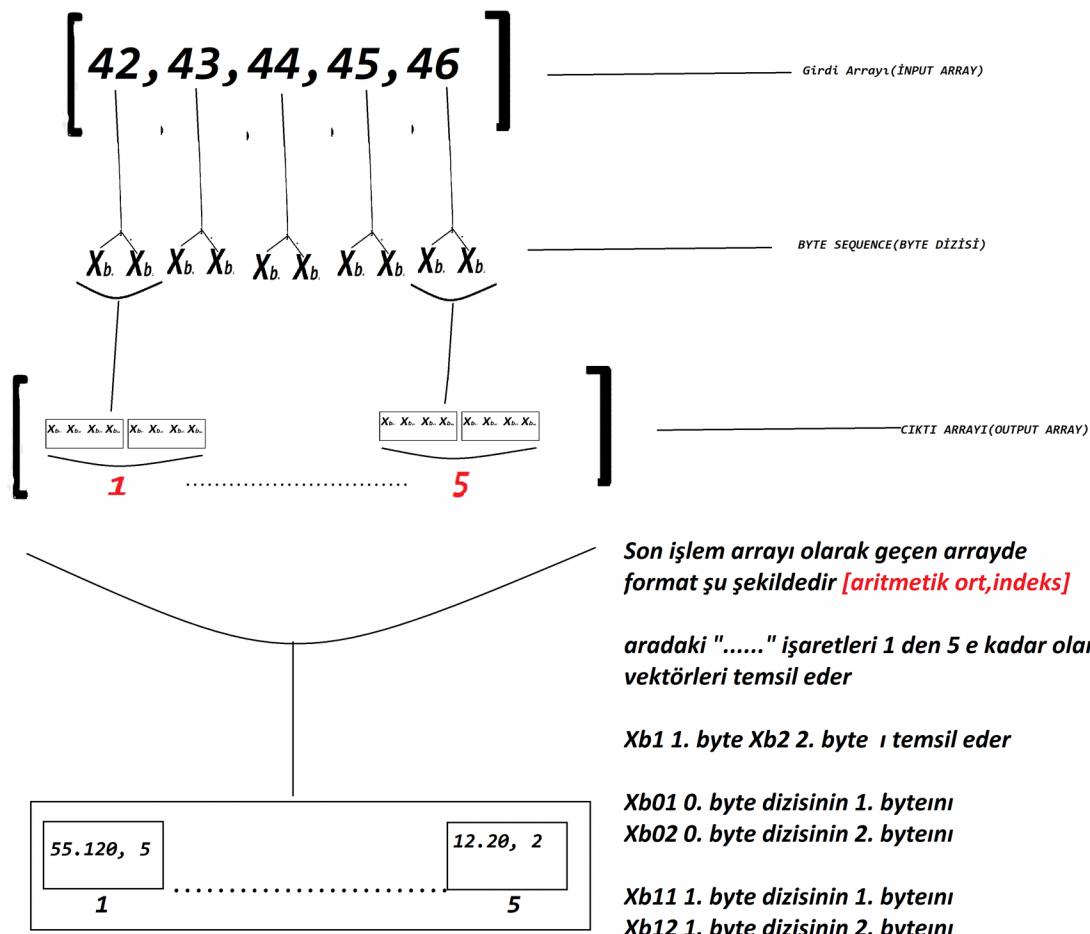
Şimdi ise açıklamalara gelelim Burada ***Input Array*** Algoritmanın üzerinde çalışacağı dizidir veya diğer bir değiş ile "**Örütü dizisi**" ***Output Array*** ise çıktıının daha özgür işlenebilmesi için oluşturulmuş döngünün sonunda bize gelen ***Matrix*** formatındaki vektör içinde vektör yapısıdır ***New array*** ise burada ***Output Array***'ı kullanıcıya okunur veri olarak vermeye yarayan değişkendir içinde gömülü bir ***For döngüsü*** vardır ve bunun sayesinde tek satırda ***1D Array*** içinde yani bir boyutlu bir array içinde tüm çıktıyı bize verebilir. Son olarak ise ***nQ*** değişkeni var buda temel unsurlar arasında yer alır ve test verisinin aralığını-

büyüklüğünü belirler ayrıca grafik eksenindeki aX aY eksenlerinin ortalanmasında da yardımcı olmak için verilmiş bir değerdir manuel sayı girmektense değişken ile kodu daha okunur hale getirmeyi hedefler.

Şimdi ise Değişken unsurların açıklamalarına gelelim Burada iki tane var bu unsurlardan birisi **Qout** idir Qout dinamik değişkeni her seferinde **Input array** içinden alından **Q** Liste değişkeninin içindeki her bir byte türüne çevrilip sonrasında **List()** fonksiyonu ile sayısal listeye çevirilmiş değerleri tekrar okuyup ikinci kademedede tekrar byte düzeyinde işleyip daha uzun bir **Byte sequence** Yani **Byte dizisi** elde edersiniz buradaki amaç aslında buradaki sayısal verilerin matematiksel birleşimlerinden çıkan değerlerin karşılaştırılarak bir sıralama elde edilmesine yardımcı olmak için bu değerleri elde etmeye yaranan değişkendir. **Q** ise burada **Input Array** den gelen sayısal verilerin stringe çevirilerek encode edilmesi ve **List()** fonksiyonu ile **Byte Sequence** yani **Byte dizisi** ne çevrilmesinde ve sonrasında bunların **Qout** gibi değişkenler içerisinde işlenmesine olanak tanıyan değişkendir.

Kısaca bu **temel** ve **Değişken** unsurlar bu şekilde açıklanabilir zaten algortima yukarıda verilen değişkenlerin üzerine kuruludur şimdi ise algoritmanın neden **Deterministik** olduğuna?

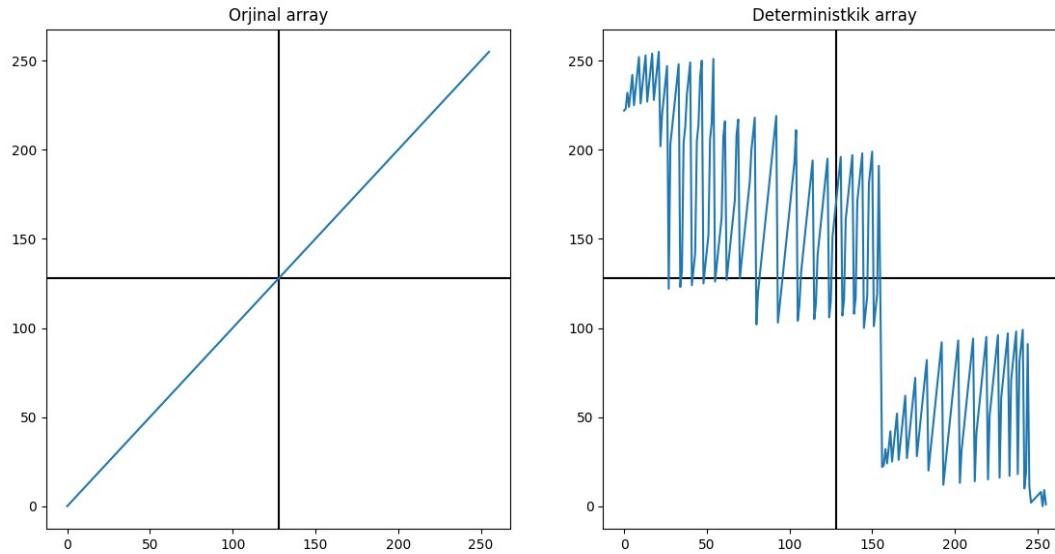
Bu algoritmada amaç kısaca şudur; aşağıda kısaca bir şema ile açıklamaya çalışacağım



Yukarıdaki bilgilere ve şemadada belirtildiği gibi "Son işlem arrayi" en sondaki arraydir ve bu array aritmetik ortalamalara göre sıralanır aritmetik ortalamalar ise vektörlerin içindeki iki kademeli byte vektörlerinden elde edilir yani o yüzden rastgele gibi durur ancak aslında mantığı bu şekildedir yani random bir şey üretmiyor.

Bu algoritma kısaca bu şekilde açıklanabilir benim tarafımdan geliştirildi ve muhtemelen geliştirilmeye devam edecek

Ve son olarak iki arrayin grafiğini verelim



Test edebilmeniz için ham veriler;

Orjinal array = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255]

Deterministik Array = [222, 223, 232, 224, 233, 242, 225, 234, 243, 252, 226, 235, 244, 253, 227, 236, 245, 254, 228, 237, 246, 255, 202, 220, 229, 238, 247, 122, 203, 212, 221, 230, 239, 248, 123, 132, 204, 213, 231, 240, 249, 124, 133, 142, 205, 214, 241, 250, 125, 134, 143, 152, 206, 215, 251, 126, 135, 144, 153, 162, 207, 216, 127, 136, 145, 154, 163, 172, 208, 217, 128, 137, 146, 155, 164, 173, 182, 200, 209, 218, 102, 120, 129, 138, 147, 156, 165, 174, 183, 192, 201, 210, 219, 103, 112, 121, 130, 139, 148, 157, 166, 175, 184, 193, 211, 104, 113, 131, 140, 149, 158, 167, 176, 185, 194, 105, 114, 141, 150, 159, 168, 177, 186, 195, 106, 115, 151, 160, 169, 178, 187, 196, 107, 116, 161, 170, 179, 188, 197, 108, 117, 171, 180, 189, 198, 100, 109, 118, 181, 190, 199, 101, 110, 119, 191, 111, 22, 23, 32, 24, 33, 42, 25, 34, 43, 52, 26, 35, 44, 53, 62, 27, 36, 45, 54, 63, 72, 28, 37, 46, 55, 64, 73, 82, 20, 29, 38, 47, 56, 65, 74, 83, 92, 12, 21, 30, 39, 48, 57, 66, 75, 84, 93, 13, 31, 40, 49, 58, 67, 76, 85, 94, 14, 41, 50, 59, 68, 77, 86, 95, 15, 51, 60, 69, 78, 87, 96, 16, 61, 70, 79, 88, 97, 17, 71, 80, 89, 98, 18, 81, 90, 99, 10, 19, 91, 11, 2, 3, 4, 5, 6,

7, 8, 0, 9, 1]
