



# Sort Alqoritmləri

Sorting Algorithms

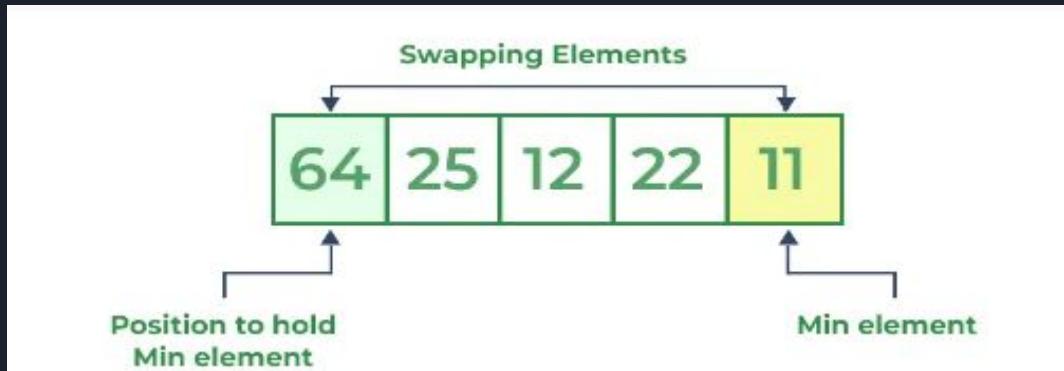


# Sort alqoritmi nədir?

Sort alqoritmi verilmiş massivdə və ya elementlər ardıcılılığında elementlər üzərində müqayisə operatorundan istifadə edərək elementləri düzgün sıralandıran bir alqoritmdir. Random sıralanan bu ədədləri müəyyən alqoritmlərdən istifadə edərək çeşidləmək mümkündür. Sort alqoritminin bir neçə növü vardır. Biz əsas 6-dan danışacıq. Selection Sort, Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, Heap Sort. Başlayaq Selectiondan.

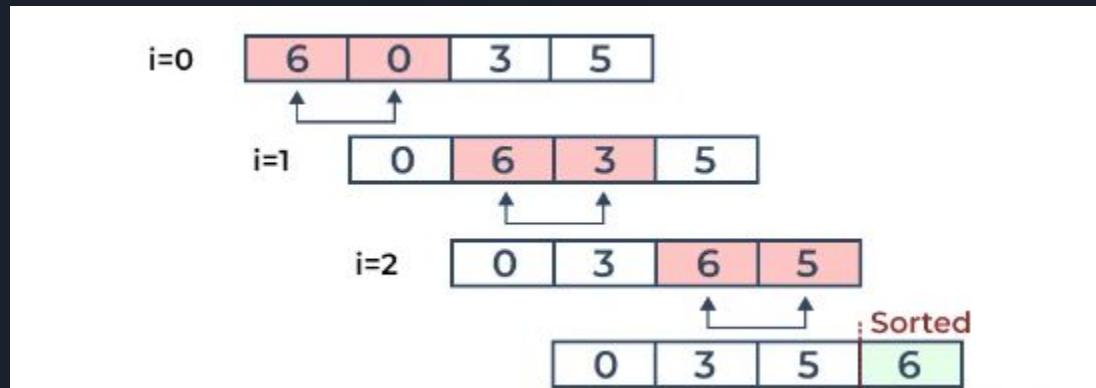
# Selection Sort

Seçmə üsulu çeşidlənməmiş siyahıdan hər hansı ən kiçik ədədi (və ya ən böyük) götürür və ona görə yerdəyişmə edir. Alqoritm ən kiçik ədədi tapır sonra onu birinci ədədlə əvəz edir. Növbəti addımda ən kiçik ikinci ədədi tapır və ikinci indekslə əvəz edir və bu ardıcılıqla massiv çeşidlənir.



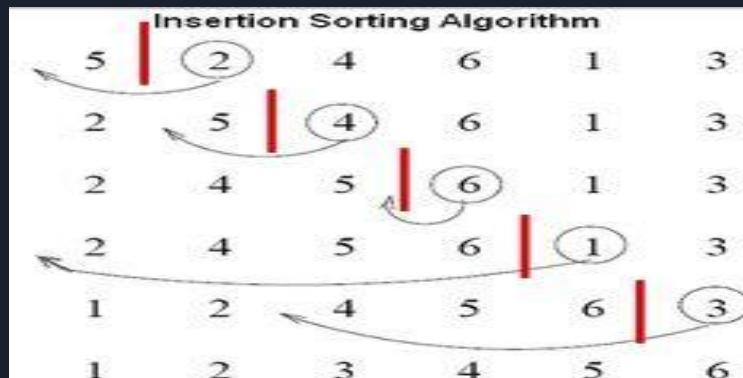
# Bubble Sort

Bubble Sort yanaşı ədədləri müqayisə edərək onları uyğun qaydada yerdəyişmə edir. Bu alqoritm böyük məlumatlarla işləmək üçün uyğun deyil çünki, onun avarage və worst case olduqca yüksəkdir. Şəkildə də göründüyü kimi 6 və 0 müqayisə olunur, yerdəyişmə edilir və alqoritm təkrarlanır.



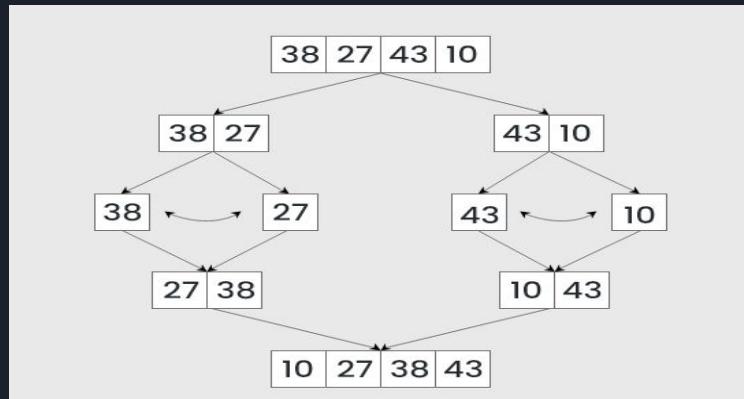
# Insertion Sort

Insertion Sortda massiv faktiki olaraq çeşidlənmiş və çeşidlənməmiş hissəyə bölünür. Çeşidlənməmiş hissədən dəyərlər seçilir və çeşidlənmiş hissədə düzgün mövqeyə yerləşdirilir. İkinci elementdən başlayaraq özündən əvvəlki elementlərlə özünü qarşılaşdıraraq , böyük elementi sıradan sağa doğru yerləşməsi prinsipinə əsaslanır.



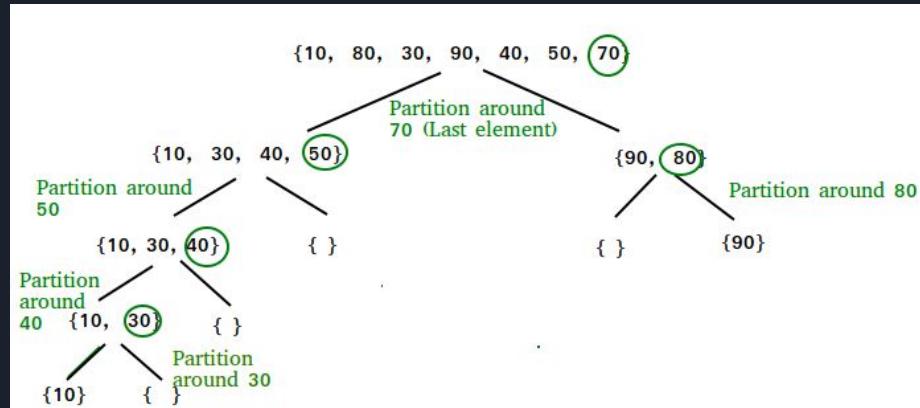
# Merge Sort

Merge Sortda massiv daha kiçik alt massivlərə bölünür, alt massivlər çeşidlənir və onları birləşdirərək son çeşidlənmiş massivi yaratmaqla işləyən bir alqoritmdir. Daha aydın desək massiv yarıya bölünür, hər yarısı çeşidlənir və birləşdirilir. Bütün massiv sıralanana qədər bu proses təkrarlanır.



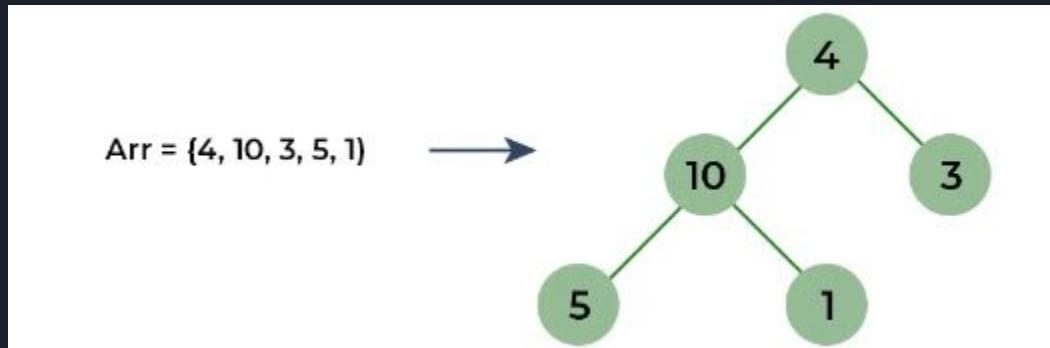
# Quick Sort

Massivdə bir element pivot kimi seçir və sonra verilmiş massivi seçilmiş pivot elementi ətrafında bölür. QuickSortda böyük massiv iki massivə bölünür ki, onlardan biri Pivotdan kiçik dəyərləri, digər massiv isə pivotdan böyük olan dəyərləri saxlayır. Sonra alınan sol və sağ massiv özü də bu prosesi davam etdirir. Bu alqoritm böl və fəth et məntiqi ilə çalışır.



# Heap Sort

Heap sort məşhur və səmərəli bir çeşidləmə alqoritmidir. Heap sort massivin elementlərini yığın adlanan xüsusi növ tam ikili ağac kimi vizuallaşdırmaqla işləyir. Bu sortdan istifadə etmək üçün massivlər və ağaclar haqqında bilik lazımdır. Bu, ilk olaraq minimum elementi tapdığımız və başlanğıcda yerləşdirdiyimiz Selection Sorta bənzəyir.





# Zaman mürəkkəbliyi

	Best Case	Average Case	Worst Case
Selection Sort	$n$	$n^2$	$n^2$
Bubble Sort	$n$	$n^2$	$n^2$
Insertion Sort	$n$	$n^2$	$n^2$
Merge Sort	$n \log n$	$n \log n$	$n \log n$
Quick Sort	$n \log n$	$n \log n$	$n^2$
Heap Sort	$n \log n$	$n \log n$	$n \log n$



Bu alqoritmlərin bəzilərinin işləndiyi kod  
örnəkləri:

```
int[] arr = new int[5] { 8, 5, 7, 3, 1 };
int n = 5, i, j, val;
Console.WriteLine("Insertion Sort");
Console.Write("Initial array is: ");
for (i = 0; i < n; i++){
    Console.Write(arr[i] + " ");
}
for (i = 1; i < n; i++){
    val = arr[i];
    for (j = i - 1; j >= 0;){
        if (val < arr[j]){
            arr[j + 1] = arr[j];
            j--;
            arr[j + 1] = val;
        }
        else{
            break;
        }
    }
    Console.Write("\nSorted Array is: ");
    for (i = 0; i < n; i++){
        Console.Write(arr[i] + " ");
    }
}
```

## Insertion Sort

# Selection Sort

```
int[] arr = { 4, 5, 2, 3, 1, 7 };
int n = arr.Length;
for (int i = 0; i < n - 1; i++)
{
    int min_idx = i;
    for (int j = i + 1; j < n; j++) {
        if (arr[j] < arr[min_idx])
        {
            min_idx = j;
        }
    }
    int temp = arr[min_idx];
    arr[min_idx] = arr[i];
    arr[i] = temp;
}
foreach (int item in arr)
{
    Console.WriteLine(item);
}
```

# Bubble Sort

```
int[] arr = { 78, 55, 45, 98, 13 };
int temp;
for (int j = 0; j <= arr.Length - 2; j++)
{
    for (int i = 0; i <= arr.Length - 2; i++)
    {
        if (arr[i] > arr[i + 1])
        {
            temp = arr[i + 1];
            arr[i + 1] = arr[i];
            arr[i] = temp;
        }
    }
}
Console.WriteLine("Sorted:");
foreach (int p in arr)
    Console.Write(p + " ");
```