Linear complexity’de, elimizdeki veri seti arttıkça, çalıştırılma süreside doğru orantılı şekilde artış gösterir.

O(N²) — Quadratic Complexity Time Algorithm

Quadratic Complexity input büyüklüğünün karesi ile doğru orantılıdır.

O(log N) — Logarithmic Complexity

Logarithmic time complexity, genelde her seferinde problemi ikiye bölen algoritmalarda kullanılır. (kitabda z herfiyle

nese tapmaq)

· Selection Sort+

· Bubble Sort+

· Insertion Sort+

· Merge Sort+

· Quick Sort+

· ShellSort+

· Iterative Merge Sort

· Iterative Quick Sort

· Heap Sort+

· Counting Sort

· Radix Sort+

· Bucket Sort

· TimSort

· Comb Sort

· Pigeonhole Sort

· Cycle Sort-

· Cocktail Sort

· Strand Sort

· Bitonic Sort

· Pancake sorting--

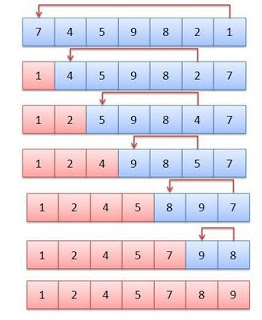
· Tree Sort

Sorting Algorithms are methods of reorganizing a large number of items into some specific order such as highest to

lowest, or vice-versa, or even in some

alphabetical order.

Selection Sort -



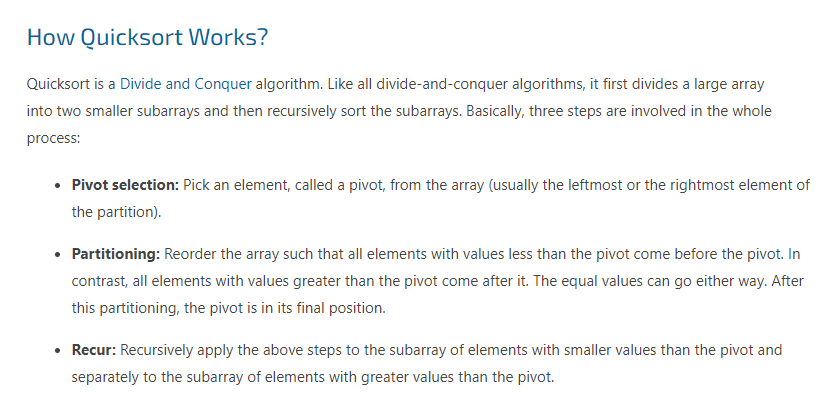
İlk elemendan son elemana kadar her elemanı kendinden sonra gelen kendinden küçük elemanların en küçüğü ile yer değiştiriyoruz. Algoritma aşağıdaki gibi çalışır:

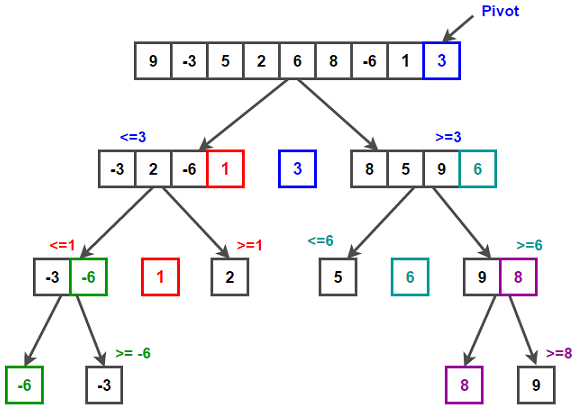
Listedeki en küçük değerli öğeyi bul.

İlk konumdaki öğeyle bulunan en küçük değerli öğenin yerini değiştir.

Yukarıdaki adımları listenin ilk elemanından sonrası için (ikinci elemandan başlayarak) yinele.

Quick Sort – en sagdaki pivota gore qruplasdiririq





Verinin hafızada sıralı tutulması için geliştirilen sıralama algoritmalarından (sorting algorithms) bir tanesidir.

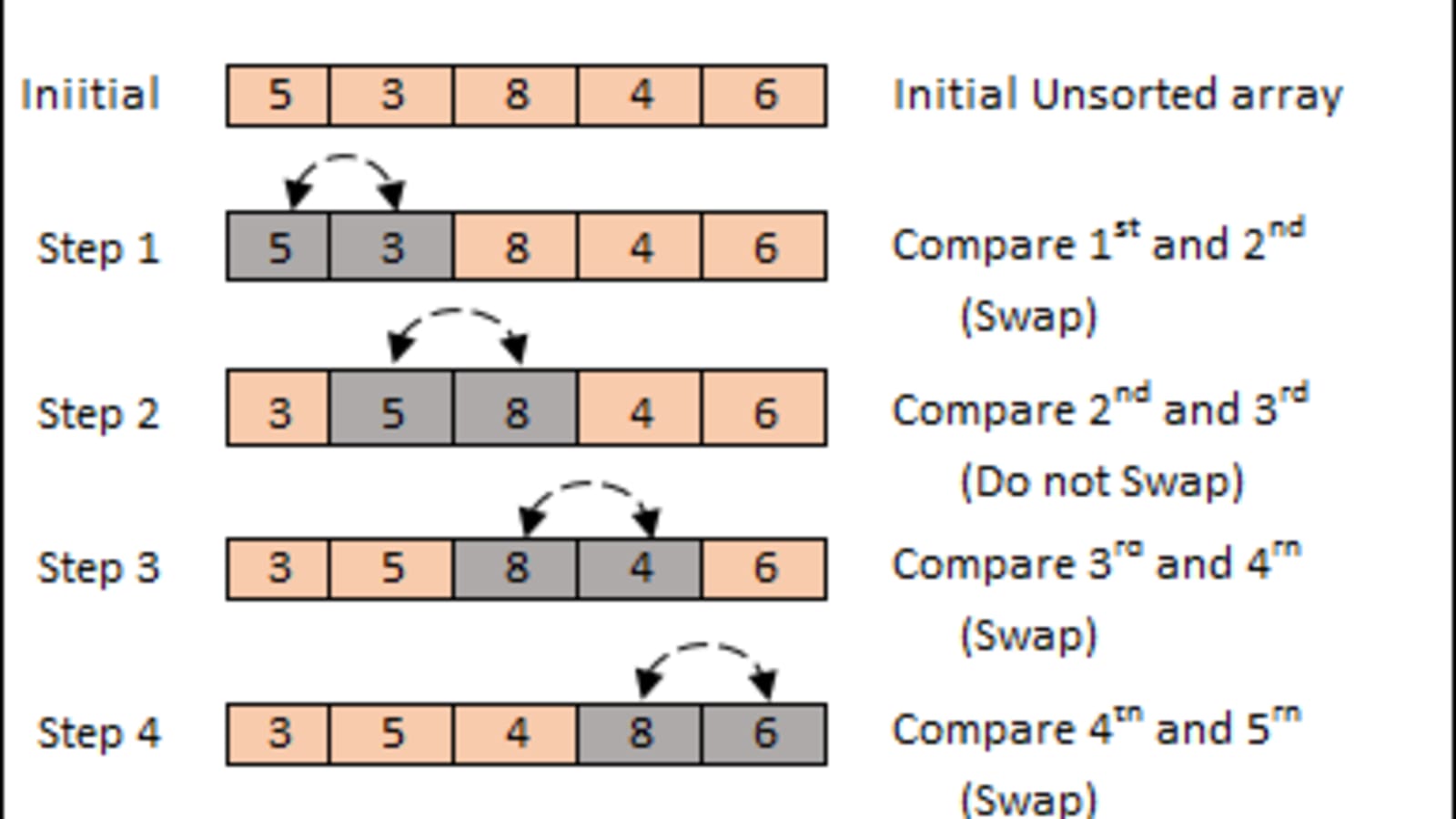
Basitçe sıralanacak olan dizideki orta noktada (mean) bulunan bir sayıyı seçerek diğer bütün sayıları bu orta

sayıdan büyük veya küçük diye sınıflayarak sıralama yapmayı hedeflemektedir. Bu açıdan bir parçala fethet

(divide and conquere) yaklaşımıdır. Ayrıca bu seçilen orta noktaya eksen (pivot) adı da verilir çünkü bütün

diğer sayılar bu sayının ekseninde sıralanacaktır.

Bubble sort- addim addim yandakiyla muqayise edir

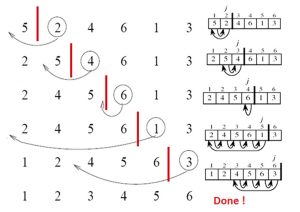


Bubble sort, en basit sıralama algoritmalarından biridir. Karşılaştırma temelli olan bu algoritmada,

listedeki her bir eleman yanındaki eleman ile karşılaştırılır. Eğer ilk elemanın değeri, ikinci elemanın

değerinden büyükse, iki eleman yer değiştirir. Daha sonra ikinci ve üçüncü elemanların değerleri karşılaştırılır.

Insertion sort – burda bubbledan ferqli olaraq yandakiyla swap elemir, en basha kecirdir kicik olani



Merge Sort – evvelce boluruk, siraliyiriq sonra da evvelki kimi birlesdiririk



Birleştirmeli sıralama algoritmamız da buna benzer bir mantık kullanıyor. Diziyi ikiye bölüp oluşan alt kümeleri tekrar birleştirmeli sıralama algoritmasına alıyor ve böylece dizi sürekli ikiye bölünmüş oluyor. Elemanları ikişerli olarak karşılaştırıp sıralı biçimde birleştiriyor, daha sonra bu iki elemanlı alt kümeleri sıralı biçimde birleştirerek dört elemanlı alt kümeler elde ediyor. Böyle devam ettiğimizde, örneğin dizimiz 16 elemanlı ise elimizde 8 elemanlı, elemanları sıralı iki alt küme kalıyor. Bu alt kümeleri de aynı şekilde sıralı biçimde birleştirdiğimiz zaman dizimiz sıralanmış oluyor. Aşağıdaki resimde bu işlemleri daha net görebiliyoruz.

Shell Sort - <http://cagataykiziltan.net/algoritmalar/1-siralama-algoritmalari/7-shell-siralama-shell-sort/>

Heap Sort-agac kimi dusunuruk, en tepede en boyuk eded olur. Sonra onu ixtisar edirik en tepeye diger en boyuk eded kecir, bele bele davam edir. Ixtisar etdikce sirayla yaziriq ve azalan sirayla duzmush oluruq.

http://bilgisayarkavramlari.com/2008/08/09/yiginlama-siralamasi-heap-sort/

Radix Sort – ededleri xanalrina gore yaziriq alt-alta. Sonra 3cu xanani azdan coxa siraliyiriq(asagdan yuxari). Sonra 2ci xanani sonra 1ci xanani ve belece ededler azdan coxa dogru siralanmis olur.

Counting Sort- nece dene eded varsa 0dan baslayib en cox eded