Veri Yapıları ve Algoritmalar

Sıralama Algoritmaları

Öğr. Gör. M. Ozan AKI

r1.0

Sıralama Algoritmaları

Bilgisayar ortamında tutulan bilgiler çoğunlukla mantıksal bir bütünlüğü olan veri kümlerinden oluşur.
Bu veri kümeleri üzerinde (hızlı) işlem (ekleme, silme, düzenleme..) yapabilmek için bu verilerin belirli kriter ya da kriterlere göre sıralanmışolması gerekir.
Sıralama Algoritmaları, herhangi veri modeli şeklinde saklanan veri kümelerini istenen kritere göre sıralayan algoritmalardır.

Sıralama Algoritmaları

En çok kullanılan belli başlı sıralama algoritmaları şunlardır;

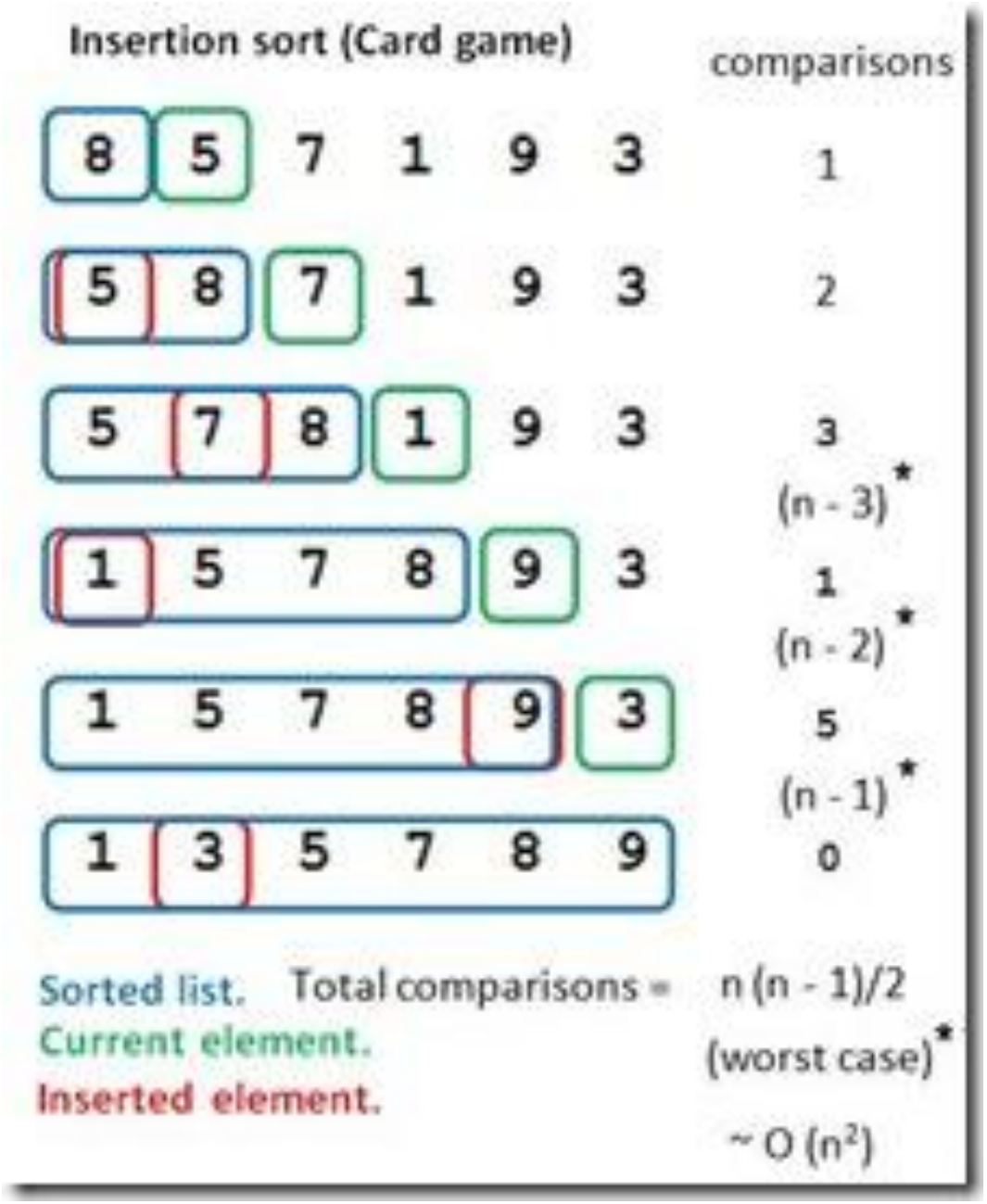
- Insertion Sort (Araya Ekleme Sıralaması)
- Selection Sort (Seçmeli Sıralama)
- Bubble Sort (Kabarcık Sıralaması)
- Merge Sort (Birleşmeli Sıralama)
- Quick Sort (Hızlı Sıralama)

Insertion Sort

Bu sıralama algoritmasında, sıralanacak dizi iki parçalı düşünülür. İlk parça, iki elemandan başlayarak dizi boyutuna kadar büyürken diğer kısım dizi boyutundan küçülerek yok olur.

Her iterasyonda, sağ parçadan bir eleman alınarak sol parçadaki uygun yere konulur. Bu sıralama algoritmasında dizide araya sokma işlemi vardır. Bunun için, uygun yer bulunana kadar dizi elemanları ötelenmek durumundadır.

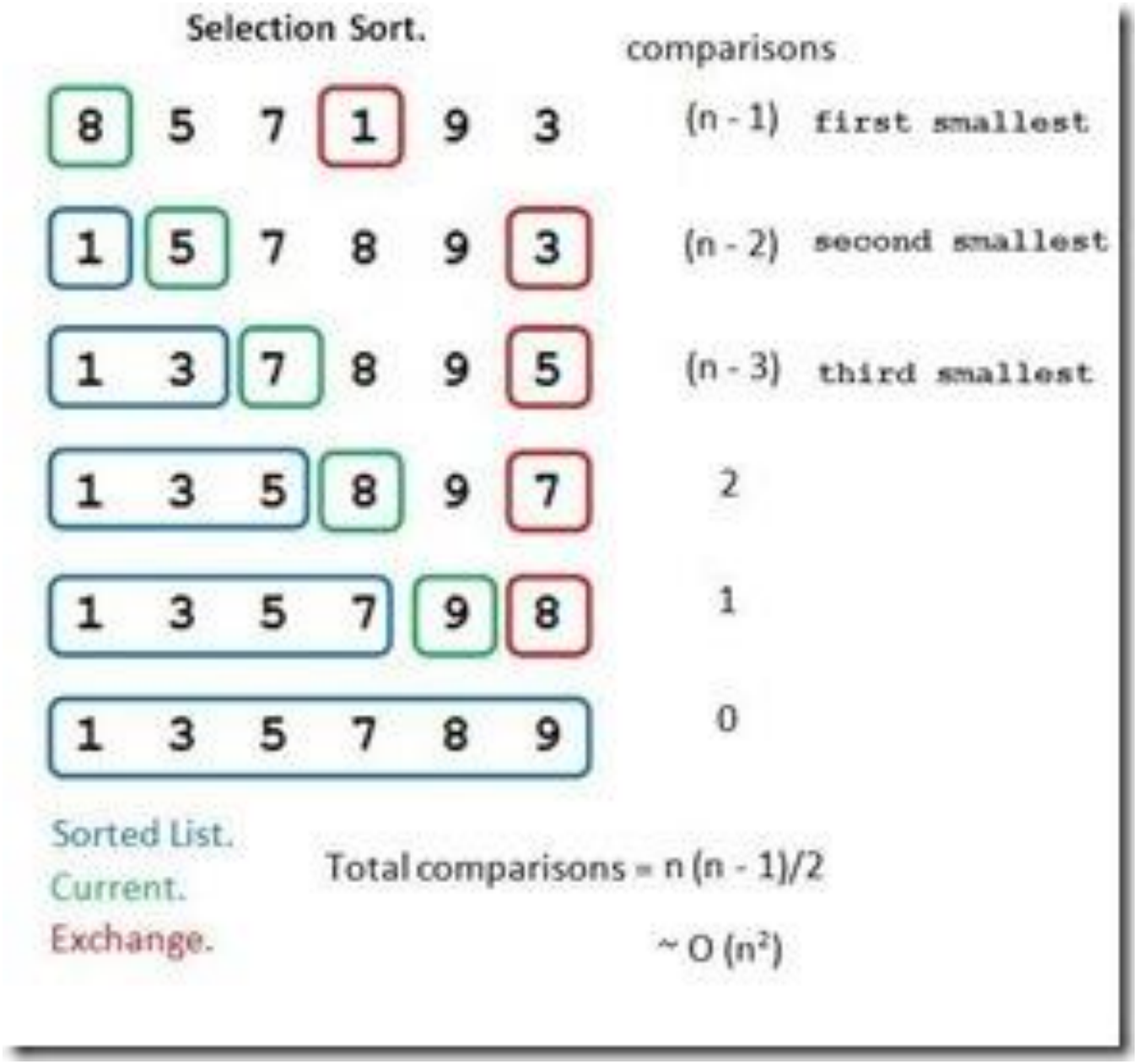
Insertion Sort



Selection Sort

Bu sıralama algoritmasında, işlem dizinin bir ucundan başlar. İlk eleman alınır. Dizinin kalan elemanları içerisinde en küçük eleman bulunur ve ilk eleman ile yer değiştirir.
Daha sonra ikinci eleman alınır ve kalan elemanlar içerisinde en küçük eleman aranır. Bulunduğunda ikinci eleman ile yer değiştirilir. İşlem bu şekilde tüm dizi sonlanana kadar devam eder.

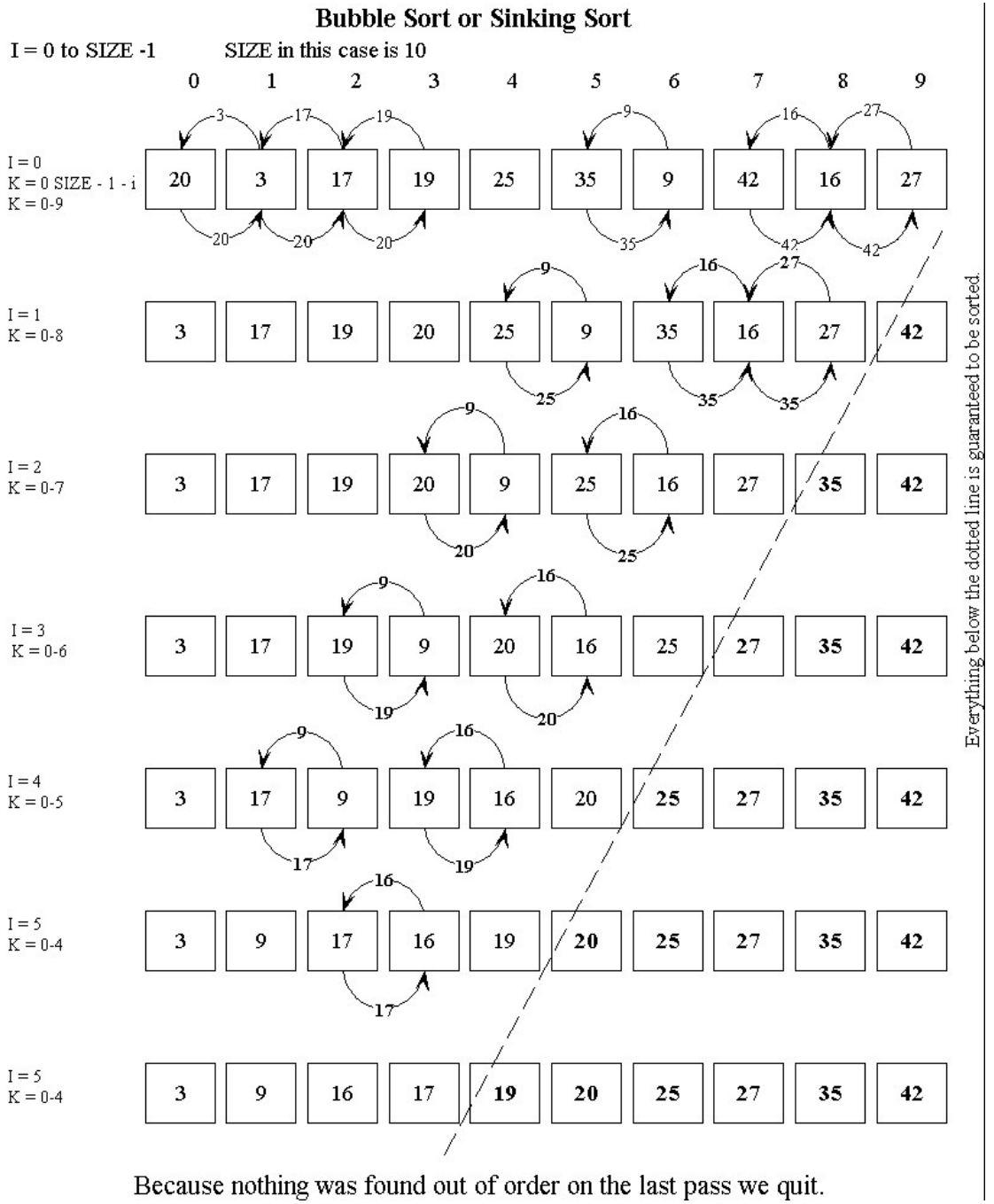
Selection Sort



Bubble Sort

Bu sıralama algoritmasında, dizi üzerinde bir uçtan diğer uca gidilirken sadece komşu iki eleman eğer sıralamaya uygun değilse yer değiştirilir, uygunsa olduğu gibi bırakılır. Ancak her geçişten sonra halen değişmesi gereken elemanlar kalabileceğinden, işlem, iç döngüsü her seferinde bir eksik çevrim yapan iç içe iki döngü ile sağlanır.

Bubble Sort


Bubble Sort or Sinking Sort
I = 0 to SIZE -1
SIZE in this case is 10
0 1 2 3 4 5 6 7 8 9
I = 0
K = 0 SIZE - 1 - i
K = 0-9
20 3 17 19 25 35 9 42 16 27
I = 1
K = 0-8
3 17 19 20 25 9 35 16 27 42
I = 2
K = 0-7
3 17 19 20 9 25 16 27 35 42
I = 3
K = 0-6
3 17 19 9 20 16 25 27 35 42
I = 4
K = 0-5
3 17 9 19 16 20 25 27 35 42
I = 5
K = 0-4
3 9 17 16 19 20 25 27 35 42
I = 5
K = 0-4
3 9 16 17 19 20 25 27 35 42
Everything below the dotted line is guaranteed to be sorted.
Because nothing was found out of order on the last pass we quit.

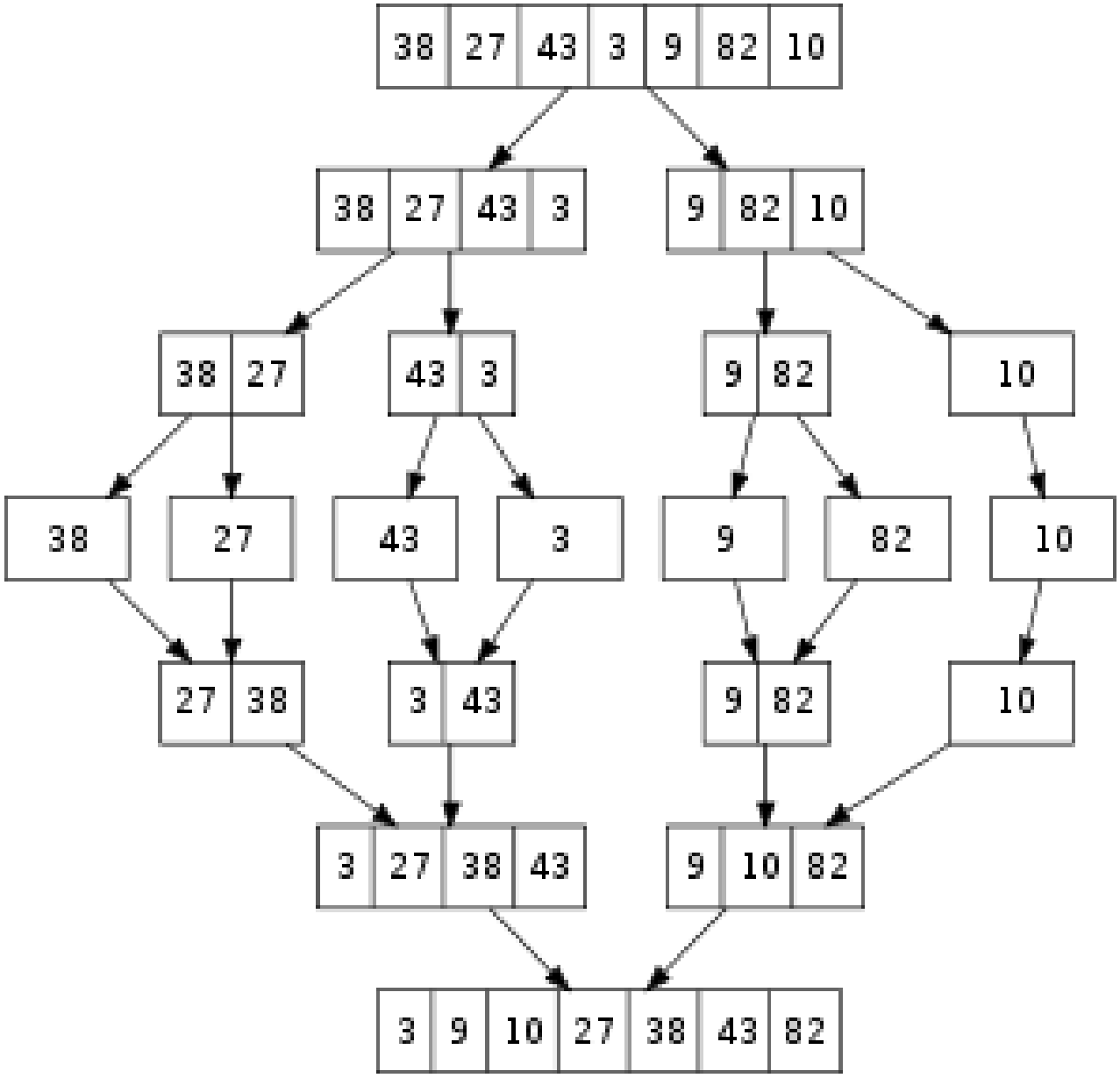
Merge Sort

Bu sıralama algoritması recursive (yinelemeli) bir yapıya sahiptir. Dizi, her bir kümede tek eleman kalıncaya kadar her seferinde ikiye bölünerek yinelenir.

Bu aşamadan sonra yinelemeli fonksiyonlar dönmeye başladığında birleştirme işlemi sırasında dönen her iki küme kendi aralarında sıralanarak birleştirilir.

Ve sonunda sıralanmış olarak yeniden bütün dizi elde edilir.

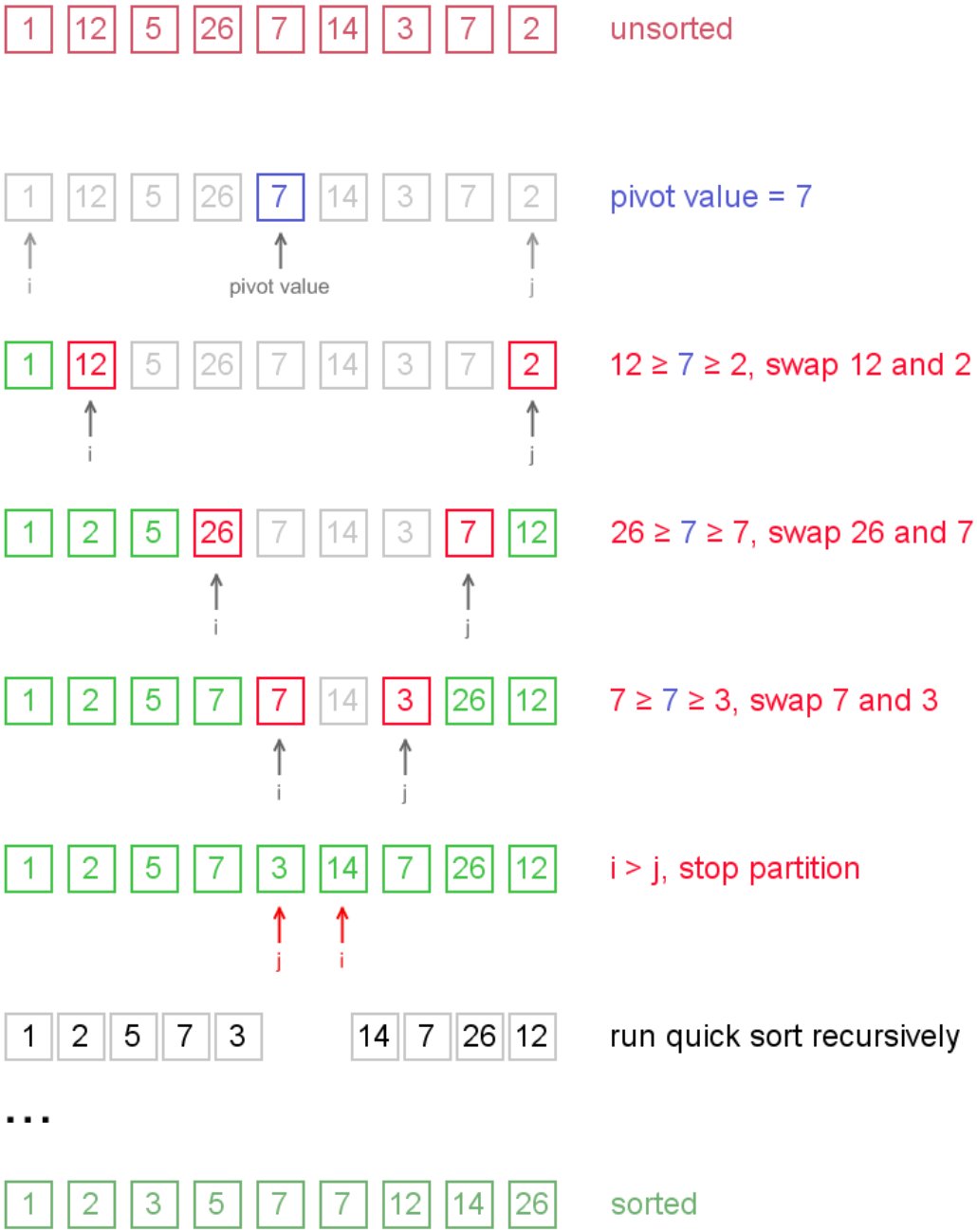
Merge Sort


38 27 43 3 9 82 10
38 27 43 3
9 82 10
38 27
43 3
9 82
10
38
27
43
3
9
82
10
27 38
3 43
9 82
10
3 27 38 43
9 10 82
3 9 10 27 38 43 82

Quick Sort

Bu sıralama algoritması recursive (yinelemeli) bir yapıya sahiptir. Dizi içerisinden seçilen herhangi sınır (pivot) değerine göre dizi bu değerden büyük olanlar ve küçük olanlara ayrılarak dizi ikiye bölünür.
İkiye bölünen bu diziler sıralama fonksiyonu kendisini tekrar çağırarak tekrar ikiye bölünür. Bölünecek dizi kalmayana kadar devam eden bu işlemin sonunda tüm dizi sıralanmış hale gelir.

Quick Sort


1 12 5 26 7 14 3 7 2
unsorted
1 12 5 26 7 14 3 7 2
i
pivot value
j
pivot value = 7
1 12 5 26 7 14 3 7 2
i
j
12 ≥ 7 ≥ 2, swap 12 and 2
1 2 5 26 7 14 3 7 12
i
j
26 ≥ 7 ≥ 7, swap 26 and 7
1 2 5 7 7 14 3 26 12
i
j
7 ≥ 7 ≥ 3, swap 7 and 3
1 2 5 7 3 14 7 26 12
j
i
i > j, stop partition
1 2 5 7 3
14 7 26 12
run quick sort recursively
...
1 2 3 5 7 7 12 14 26
sorted