

Selection Sort algoritması

Mehmet ÖZER ozermehmet911@gmail.com

Selection Sort algoritması, basit ve anlaşılır bir sıralama yöntemi olup, dizideki elemanları küçükten büyüğe doğru sıralamak için kullanılan bir algoritmadır. Temel çalışma prensibi, dizinin her bir konumuna sırasıyla o kısımdaki en küçük elemanı yerleştirmek üzerine kuruludur.

Algoritma, sıralanmamış bir dizinin ilk elemanından başlayarak, dizinin geri kalan kısmında bulunan en küçük elemanı bulur ve mevcut konumdaki eleman ile yer değiştirir. Bu işlemi dizinin sonuna kadar tekrarlar. Her bir iterasyonda, dizinin bir kısmı daha sıralanmış olur ve bir sonraki adımda artık o kısım dikkate alınmaz..

Algoritmanın performansı, dizinin başlangıç durumundan bağımsızdır. Zaten sıralı bir dizi bile olsa, yine aynı sayıda karşılaştırma işlemi yapacaktır. Bu, Selection Sort'un dezavantajlarından biridir. Buna karşılık, bellek kullanımı açısından oldukça verimlidir, çünkü her defasında yalnızca bir eleman için yer değişimi yapar.

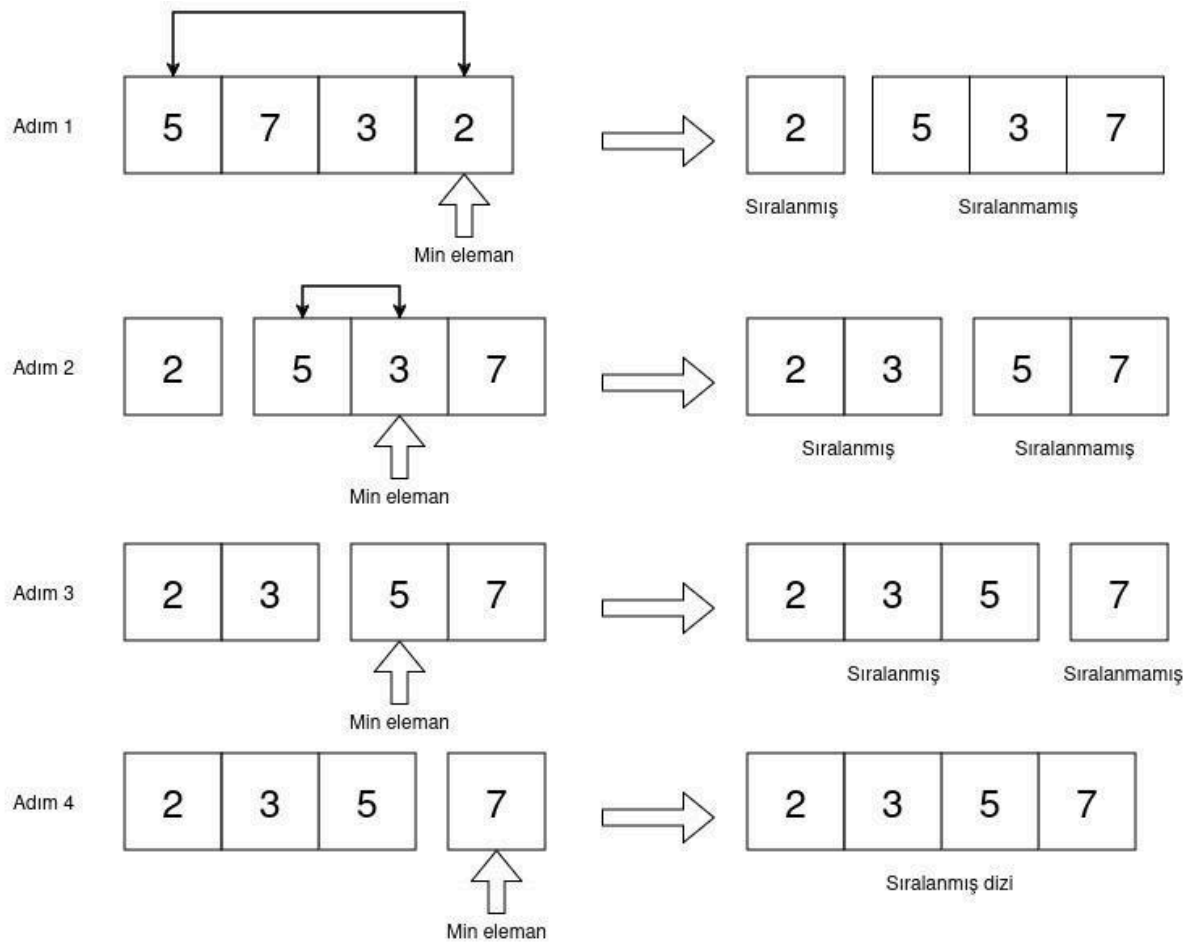
Algoritmanın Çalışma Mantığı

1. Dizinin ilk elemanı, en küçük eleman olarak kabul edilir.
2. Dizinin geri kalan kısmında bu elemandan daha küçük bir değer aranır.
3. Daha küçük bir eleman bulunduğunda, bu eleman ile mevcut konumda bulunan eleman yer değiştirilir.
4. İşlem, dizinin her bir elemanı için tekrarlanır ve her iterasyon sonunda dizinin bir kısmı sıralanmış olur.

Her döngü sonunda, sıralanmış kısmın boyutu bir eleman artar ve sıralanmamış kısmın boyutu azalır.

Algoritmanın Avantajları ve Dezavantajları

Selection Sort, küçük veri setlerinde verimli olan bir sıralama algoritmasıdır. Zaman karmaşıklığının $O(n^2)$ olması nedeniyle büyük boyutlu dizilerde performansı düşüktür. Algoritmanın en önemli özelliği, dizinin başlangıç durumundan bağımsız olarak her zaman aynı sayıda iterasyon ve karşılaştırma yapmasıdır. Bu, algoritmanın hem avantajı hem de dezavantajıdır. Bellek açısından verimli olup in-place bir algoritma özelliği gösterir, çünkü her defasında yalnızca minimum sayıda eleman değişimi gerçekleştirir.



Selection Sort Algoritmasının Seçilen Dillerde Uygulanması

Haskell

```
selectionSort :: (Ord a) => [a] -> [a]
selectionSort [] = []
selectionSort xs = minElement : selectionSort rest
  where
    minElement = findMin xs
    rest = remove minElement xs

findMin :: (Ord a) => [a] -> a
findMin [x] = x
findMin (x:xs) = min x (findMin xs)

remove :: Eq a => a -> [a] -> [a]
remove _ [] = []
remove e (x:xs)
  | e == x    = xs
  | otherwise = x : remove e xs
```

C++

```
void selectionSort(int arr[], int n) {  
    for (int i = 0; i < n - 1; ++i) {  
        int minIdx = i;  
        for (int j = i + 1; j < n; ++j) {  
            if (arr[j] < arr[minIdx]) {  
                minIdx = j;  
            }  
        }  
        swap(arr[i], arr[minIdx]);  
    }  
}
```

Python

```
def selection_sort(arr):  
    n = len(arr)  
    for i in range(n):  
        for j in range(i + 1, n):  
            if arr[j] < arr[i]:  
                arr[i], arr[j] = arr[j], arr[i]
```

Selection Sort Algoritmasının Farklı Programlama Dillerindeki Performans Karşılaştırması: Haskell, C++ ve Python

Programlama Dili	İşletim Sistemi	Donanım	Süre	Veri Seti
Haskell	Linux version 6.8.0-49-generic	Intel i5 12500h	0.000023s	10 ⁴
C++	Linux version 6.8.0-49-generic	Intel i5 12500h	0.036s	10 ⁴
Python 3.12.3	Linux version 6.8.0-49-generic	Intel i5 12500h	1.46s	10 ⁴