

Sortir

Teknik Informatika UMJ

Dosen Pengampu: Mirza Sutrisno, M.Kom.



UMJ
UNIVERSITAS
MUHAMMADIYAH
JAKARTA

Enlightening, Empowering

Selection Sort

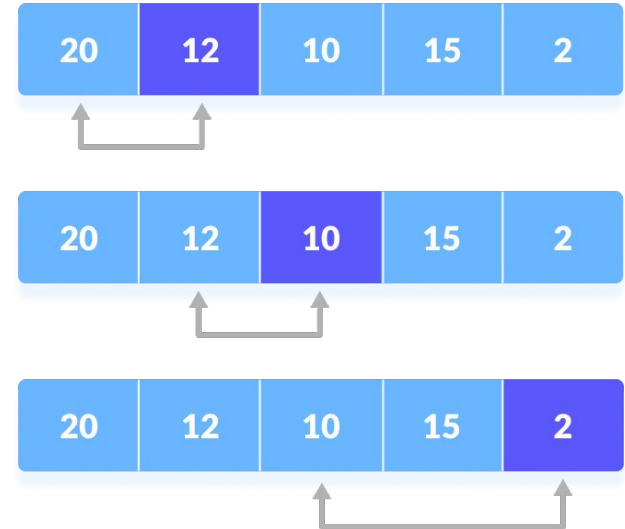
Selection sort adalah algoritma pengurutan yang memilih elemen terkecil dari daftar yang tidak disortir di setiap iterasi dan menempatkan elemen tersebut di awal daftar yang tidak disortir.

Cara Kerja

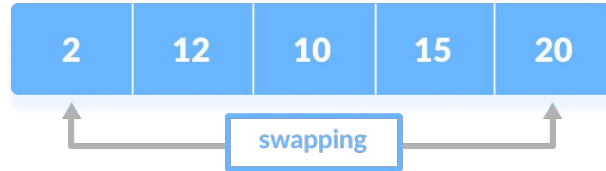
1. Tetapkan elemen pertama sebagai minimum.



2. Bandingkan minimum dengan elemen kedua. Jika elemen kedua lebih kecil dari minimum, tetapkan elemen kedua sebagai minimum. Bandingkan minimum dengan elemen ketiga. Sekali lagi, jika elemen ketiga lebih kecil, tetapkan minimum ke elemen ketiga jika tidak, jangan lakukan apa pun. Proses berlangsung hingga elemen terakhir.

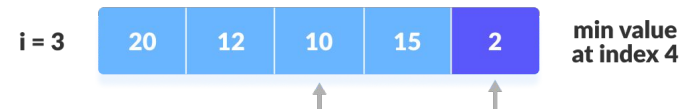
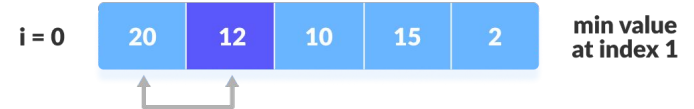


3. Setelah setiap iterasi, minimum ditempatkan di depan daftar yang tidak disortir.

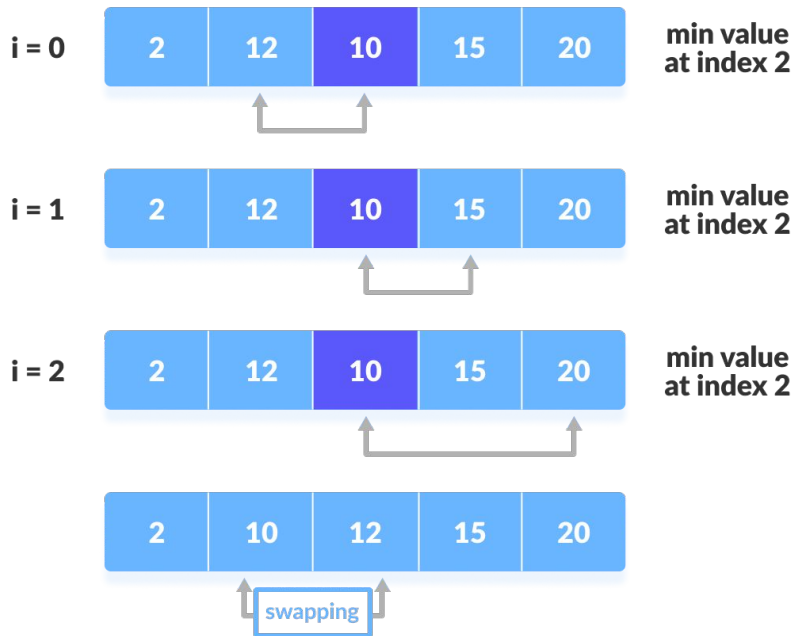


4. Untuk setiap iterasi, pengindeksan dimulai dari elemen pertama yang tidak disortir. Langkah 1 sampai 3 diulang sampai semua elemen ditempatkan pada posisi yang benar.

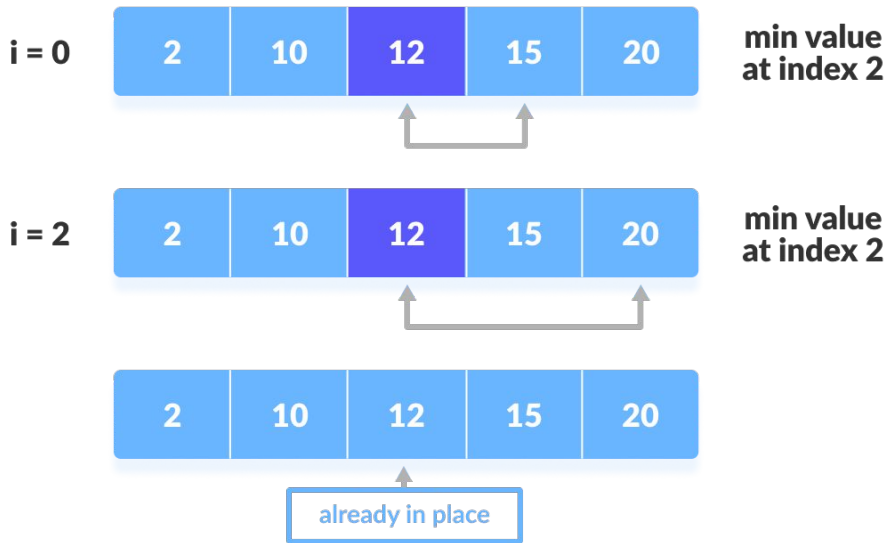
step = 0



step = 1



step = 2



Source Code

```
// Selection sort in C++
```

```
#include <iostream>
using namespace std;
```

```
// function to swap the the position of two elements
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}
```

```
// function to print an array
void printArray(int array[], int size) {
    for (int i = 0; i < size; i++) {
        cout << array[i] << " ";
    }
    cout << endl;
}
```

```
void selectionSort(int array[], int size) {
    for (int step = 0; step < size - 1; step++) {
        int min_idx = step;
        for (int i = step + 1; i < size; i++) {

            // To sort in descending order, change > to < in this line.
            // Select the minimum element in each loop.
            if (array[i] < array[min_idx])
                min_idx = i;
        }

        // put min at the correct position
        swap(&array[min_idx], &array[step]);
    }
}
```

```
// driver code
int main() {
    int data[] = {20, 12, 10, 15, 2};
    int size = sizeof(data) / sizeof(data[0]);
    selectionSort(data, size);
    cout << "Sorted array in Ascending Order:\n";
    printArray(data, size);
}
```

```
Sorted array in Ascending Order:
2 10 12 15 20

-----
Process exited after 0.0574 seconds with return value 0
Press any key to continue . . .
```

Algoritma Quicksort

Quicksort adalah algoritma pengurutan yang didasarkan pada pendekatan Divide dan Conquer.

1. Array dibagi menjadi subarray dengan memilih elemen pivot (elemen yang dipilih dari array). Saat membagi larik, elemen pivot harus diposisikan sedemikian rupa sehingga elemen yang kurang dari pivot disimpan di sisi kiri dan elemen yang lebih besar dari pivot berada di sisi kanan pivot.
2. Subarray kiri dan kanan juga dibagi menggunakan pendekatan yang sama. Proses ini berlanjut hingga setiap subarray berisi satu elemen.
3. Pada titik ini, elemen sudah diurutkan. Akhirnya, elemen digabungkan untuk membentuk array yang diurutkan.

<https://www.programiz.com/dsa/quick-sort>

Merge Sort

Merge Sort adalah salah satu algoritma sorting paling populer yang didasarkan pada prinsip Divide and Conquer Algorithm .

Di sini, sebuah masalah dibagi menjadi beberapa sub-masalah. Setiap sub-masalah diselesaikan secara individual. Akhirnya, sub-masalah digabungkan untuk membentuk solusi akhir.

<https://www.programiz.com/dsa/merge-sort>