

# Selection Sort

In this lesson, we'll cover how selection sort works and see its implementation.

We'll cover the following

- Selection sort

## Selection sort #

Maintain two parts of the array, sorted and unsorted parts. Starting with the sorted part being empty and the unsorted part being  $A[0..N - 1]$ , repeatedly find the smallest integer in the unsorted part and swap it to the end of the sorted part.

A small example will explain the process quickly. In this example, the unsorted part is bold.

- Step 0:  $[6, 3, 5, 4, 1, 2]$
- Step 1: find minimum in  $A[0..5]$  and move to 0  $\rightarrow [1, 3, 5, 4, 6, 2]$ .
- Step 2: find minimum in  $A[1..5]$  and move to 1  $\rightarrow [1, 2, 5, 4, 6, 3]$ .
- Step 3: find minimum in  $A[2..5]$  and move to 2  $\rightarrow [1, 2, 3, 4, 6, 5]$ .

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N = 6;
    int arr[N] = {6, 3, 5, 4, 1, 2};

    for (int i = 0; i < N; i++) {
        int min_idx = i;
        for(int j = i + 1; j < N; j++) {
            if (arr[j] < arr[min_idx])
                min_idx = j;
        }
        swap(arr[i], arr[min_idx]);
    }

    for (int i = 0; i < N; i++) cout << arr[i] << " "; cout << "\n";
    return 0;
}
```



In the next lesson, we'll take a look at bubble sort, another  $O(N^2)$  sorting algorithm.