

## Selection Sort

Selection sort is conceptually the simplest sorting algorithm. This algorithm will first find the **smallest** element in the array and swap it with the element in the **first** position, then it will find the **second smallest** element and swap it with the element in the **second** position, and it will keep on doing this until the entire array is sorted.

It is called selection sort because it repeatedly **selects** the next-smallest element and swaps it into the right place.

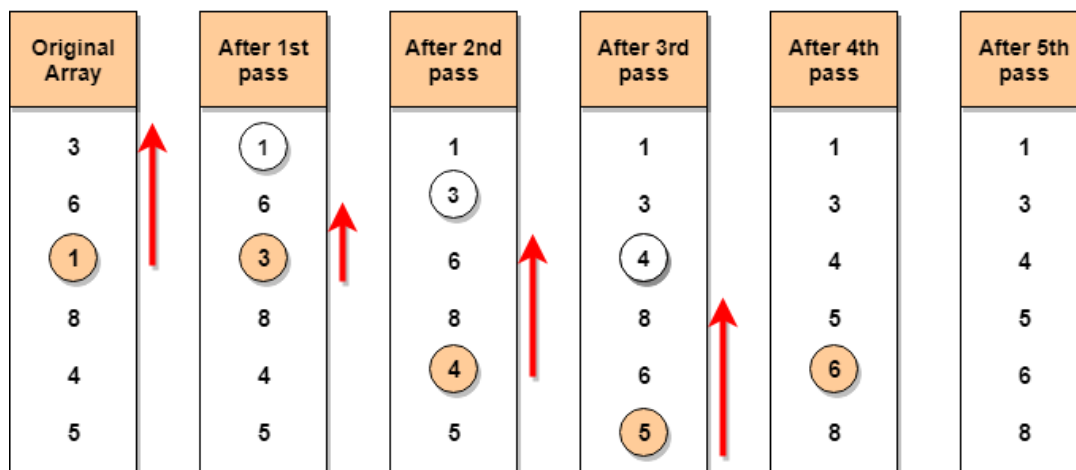
### Implementing Selection Sort Algorithm

Following are the steps involved in selection sort (for sorting a given array in ascending order):

1. Starting from the first element, we search the smallest element in the array, and replace it with the element in the first position.
2. We then move on to the second position, and look for smallest element present in the subarray, starting from index 1, till the last index.
3. We replace the element at the **second** position in the original array, or we can say at the first position in the subarray, with the second smallest element.
4. This is repeated, until the array is completely sorted.

Let's consider an array with values {3, 6, 1, 8, 4, 5}

Below, we have a pictorial representation of how selection sort will sort the given array.



In the **first** pass, the smallest element will be 1, so it will be placed at the first position.

Then leaving the first element, **next smallest** element will be searched, from the remaining elements. We will get 3 as the smallest, so it will be then placed at the second position.

Then leaving 1 and 3 (because they are at the correct position), we will search for the next smallest element from the rest of the elements and put it at third position and keep doing this until array is sorted.

To illustrate the output of the above example:

Pass 1: 1, 6, 3, 8, 4, 5

Pass 2: 1, 3, 6, 8, 4, 5

Pass 3: 1, 3, 4, 8, 6, 5

Pass 4: 1, 3, 4, 5, 6, 8

Pass 5: 1, 3, 4, 5, 6, 8

***The underlined data is the sorted list. The smallest data is determined from the unsorted list and then swapped to the first element of that unsorted list.***

Below is the code for the selection sort algorithm:

```

// function to swap elements at the given index values
void swap(int arr[], int firstIndex, int secondIndex)
{
    int temp;
    temp = arr[firstIndex];
    arr[firstIndex] = arr[secondIndex];
    arr[secondIndex] = temp;
}

// function to look for smallest element in the given subarray
int indexOfMinimum(int arr[], int startIndex, int n)
{
    int minValue = arr[startIndex];
    int minIndex = startIndex;

    for(int i = minIndex + 1; i < n; i++) {
        if(arr[i] < minValue)
        {
            minIndex = i;
            minValue = arr[i];
        }
    }
    return minIndex;
}

void selectionSort(int arr[], int n)
{
    for(int i = 0; i < n; i++)
    {
        int index = indexOfMinimum(arr, i, n);
        swap(arr, i, index);
    }
}

```

Below is a link for the selection sort tutorial:

[https://www.youtube.com/watch?time\\_continue=70&v=xWBP4IzkoyM&feature=emb\\_logo](https://www.youtube.com/watch?time_continue=70&v=xWBP4IzkoyM&feature=emb_logo)

Here is the complete output based from the sample from the video:

Given: 64 25 12 22 11

Pass 1: 11 25 12 22 64

Pass 2: 11 12 25 22 64

Pass 3: 11 12 22 25 64

Pass 4: 11 12 22 25 64