

1. Selection Sort

Algorithm

list: array of items

n: size of list

```
for i = 1 to n - 1
{
    /* set current element as minimum */
    min = i

    /* check the element to be minimum */
    for j = i+1 to n
    {
        if list[j] < list[min] then
            min = j;
    }

    /* swap the minimum element with the current element */
    if min != i then
        swap list[min] and list[i]
}
```

Time Complexity: $O(n^2)$

2. Bubble Sort

list: array of items

n: size of list

```
for i = 0 to n-1 do:
{
    swapped = false

    for j = 0 to n-1 do:
    {
        /* compare the adjacent elements */
        if list[j] > list[j+1] then
            swap( list[j], list[j+1] )
            swapped = true
    }

    /*if no number was swapped that means array is sorted now, break the loop.*/
    if(not swapped) then
        break
}
```

Time Complexity: $O(n^2)$

3. Insertion Sort

int i, key, j;

```

for (i = 1; i < n; i++)
{
    key = arr[i];
    j = i - 1;

    /* Move elements of arr[0..i-1], that are greater than key, to one position ahead of their
current position */
    while (j >= 0 && arr[j] > key)
    {
        arr[j + 1] = arr[j];
        j = j - 1;
    }
    arr[j + 1] = key;
}

```

Time Complexity: $O(n^2)$