

Insertion Sort

Insertion sort is a simple sorting algorithm that works the way we sort playing cards in our hands.

Algorithm

// Sort an arr[] of size n

insertionSort(arr, n)

Loop from i = 1 to n-1.

.....a) Pick element arr[i] and insert it into sorted sequence arr[0...i-1]

Example:

Insertion Sort Execution Example



Another Example:

12, 11, 13, 5, 6

Let us loop for i = 1 (second element of the array) to 4 (last element of the array)

i = 1. Since 11 is smaller than 12, move 12 and insert 11 before 12

11, 12, 13, 5, 6

i = 2. 13 will remain at its position as all elements in A[0..i-1] are smaller than 13

11, 12, 13, 5, 6

i = 3. 5 will move to the beginning and all other elements from 11 to 13 will move one position ahead of their current position.

5, 11, 12, 13, 6

i = 4. 6 will move to position after 5, and elements from 11 to 13 will move one position ahead of their current position.

5, 6, 11, 12, 13

```
// C++ program for insertion sort

#include <bits/stdc++.h>

using namespace std;

/* Function to sort an array using insertion sort*/

void insertionSort(int arr[], int n)
{
    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;

}

```

// A utility function to print an array of size n

```

void printArray(int arr[], int n)
{

    int i;

    for (i = 0; i < n; i++)

        cout << arr[i] << " ";

```

```
        cout << endl;

    }

/* Driver code */

int main()

{

    int arr[] = { 12, 11, 13, 5, 6 };

    int n = sizeof(arr) / sizeof(arr[0]);

    insertionSort(arr, n);

    printArray(arr, n);

    return 0;

}
```

Fun time: refer the link for better understanding

<https://youtu.be/ROalU379I3U>