

9. IMPLEMENTATION OF INSERTION SORT

Preamble

In selection sort, the smallest value among the unsorted elements of the array is selected in every pass and inserted to its appropriate position into the array. It is also the simplest algorithm. It is an in-place comparison sorting algorithm. In this algorithm, the array is divided into two parts, first is sorted part, and another one is the unsorted part. Initially, the sorted part of the array is empty, and unsorted part is the given array. Sorted part is placed at the left, while the unsorted part is placed at the right.

In selection sort, the first smallest element is selected from the unsorted array and placed at the first position. After that second smallest element is selected and placed in the second position. The process continues until the array is entirely sorted.

Steps

Step 1 - If the element is the first element, assume that it is already sorted. Return 1.

Step2 - Pick the next element, and store it separately in a key.

Step3 - Now, compare the key with all elements in the sorted array.

Step 4 - If the element in the sorted array is smaller than the current element, then move to the next element. Else, shift greater elements in the array towards the right.

Step 5 - Insert the value.

Step 6 - Repeat until the array is sorted.

Implementation in C

```
#include <stdio.h>

/* function to sort an array with insertion sort */
void insert(int a[], int n)
{
    int i, j, temp;
    for (i = 1; i < n; i++)
    {
        temp = a[i];
        j = i - 1;
```

```
/* Move the elements greater than temp to one position ahead from their
current position*/
```

```
    while(j>=0 && temp <= a[j])
    {
        a[j+1] = a[j];
        j = j-1;
    }
    a[j+1] = temp;
}
}
```

```
void printArr(int a[], int n) /* function to print the array */
```

```
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
}
```

```
int main()
```

```
{
    int a[] = { 12, 31, 25, 8, 32, 17 };
    int n = sizeof(a) / sizeof(a[0]);
    printf("Before sorting array elements are - \n");
    printArr(a, n);
    insert(a, n);
    printf("\nAfter sorting array elements are - \n");
    printArr(a, n);
    return 0;
}
```

Sample Input and Output

```
Before sorting array elements are -  
12 31 25 8 32 17  
After sorting array elements are -  
8 12 17 25 31 32
```