|  |
| --- |
| Experiment No.1 |
| Insertion Sort |
| Date of Performance: |
| Date of Submission: |

**Title**: Insertion Sort

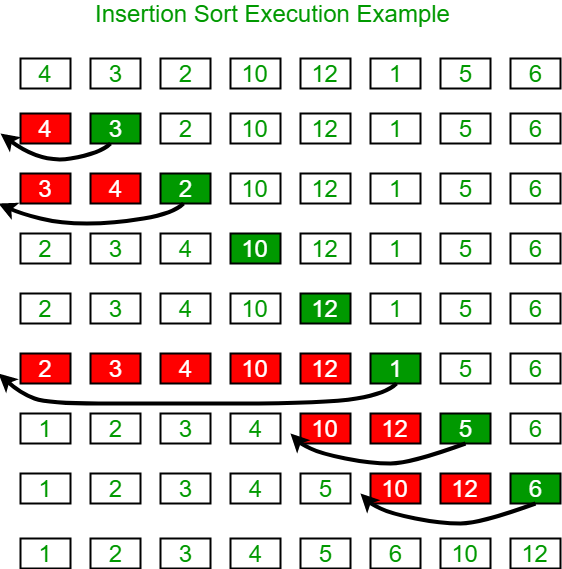
**Aim**: To implement Selection Comparative analysis for large values of ‘n’

**Objective:** To introduce the methods of designing and analysing algorithms

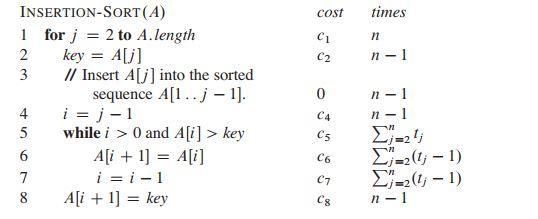
**Theory**:

Insertion sort is a simple sorting algorithm that works similar to the way you sort playing cards in your hands. The array is virtually split into a sorted and an unsorted part. Values from the unsorted part are picked and placed at the correct position in the sorted part.

**Example:**



**Algorithm and Complexity:**



**Implementation:**

#include<stdio.h>

#include<conio.h>

int main(){

int i,j,key,n;

int A[100];

clrscr();

printf("\*\*\*INSERTION SORT\*\*\*");

printf("\nEnter the size of array :");

scanf("%d",&n);

printf("\nEnter the elements: \n");

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

scanf("%d",&A[i]);

}

for(j=1;j<=n;j++){

key=A[j];

i=j-1;

while(i>0 && A[i]>key){

A[i+1]=A[i];

i=i-1;

}

A[i+1]=key;

}

printf("\nElements after sorting :");

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

printf("\n%d",A[i]);

}

return 0;

}

**Output:**



**Conclusion:** The implementation of the insertion sort algorithm demonstrated its effectiveness in sorting small to moderate-sized datasets. While its simplicity and efficiency are notable, scalability limitations highlight the need for alternative algorithms for larger datasets. Nonetheless, insertion sort remains a valuable foundational concept in computer science education.