

**AIM:** Sort a given set of N integer elements using **Selection Sort** technique and compute its time taken. Run the program for different values of N and record the time taken to sort. Plot a graph of the time taken versus N using MS Excel. The program should allow both manual entry of the array elements and also reading of array elements using random number generator. Note: In the record book students should

- Handwrite the Algorithm
- Handwrite the Program
- Pasting of the printout of the Output and Graph or Handwriting of the Output and Graph.

Note: N value should be in the range

**ALGORITHM :** sel\_sort(a[0...n-1])  
 //Sorts a given array by selection sort  
 //Input : An array a[0...n-1] of orderable elements  
 //Output : Array a[0...n-1] sorted in ascending order  
**for** i=0 to n-2 **do**  
     small\_pos=i  
     **for** j=i+1 to n-1 **do**  
         **if** a[j]<a[small\_pos]  
             small\_pos=j  
         **end if**  
     **end for**  
     swap a[i] and a[small\_pos]  
**end for**

**Program:**

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h> /* To recognise exit function when compiling with gcc*/
void selsort(int n,int a[]);

void main()
{
    int a[15000],n,i,j,ch,temp;
    clock_t start,end;

    while(1)
    {
        printf("\n1:For manual entry of N value and array elements");
        printf("\n2:To display time taken for sorting number of elements N in the range 500 to 14500");
        printf("\n3:To exit");
        printf("\nEnter your choice:");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1: printf("\nEnter the number of elements: ");
                    scanf("%d",&n);
                    printf("\nEnter array elements: ");
                    for(i=0;i<n;i++)
                    {
                        scanf("%d",&a[i]);
                    }
                    start=clock();
                    selsort(n,a);
```

```

        end=clock();
        printf("\nSorted array is: ");
        for(i=0;i<n;i++)
            printf("%d\t",a[i]);
        printf("\n Time taken to sort %d numbers is %f Secs",n, (((double)(end-
start))/CLOCKS_PER_SEC));
        break;
    case 2:
        n=500;
        while(n<=14500) {
            for(i=0;i<n;i++)
                {
                    //a[i]=random(1000);
                    a[i]=n-i;
                }
            start=clock();
            selsort(n,a);
            //Dummy loop to create delay
            for(j=0;j<500000;j++){ temp=38/600;}
            end=clock();
            printf("\n Time taken to sort %d numbers is %f Secs",n, (((double)(end-
start))/CLOCKS_PER_SEC));
            n=n+1000;
        }
        break;
    case 3: exit(0);
    }
    getchar();
}
}

```

```

void selsort(int n,int a[])
{
    int i,j,t,small,pos;
    for(i=0;i<n-1;i++)
    {
        pos=i;
        small=a[i];
        for(j=i+1;j<n;j++)
        {
            if(a[j]<small)
            {
                small=a[j];
                pos=j;
            }
        }
        t=a[i];
        a[i]=a[pos];
        a[pos]=t;
    }
}

```

OUTPUT:-

```
D:\1bm22cs195ada\selections  X  +  v

1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 14500
3:To exit
Enter your choice:1

Enter the number of elements: 4

Enter array elements: 10 9 7 4

Sorted array is: 4      7      9      10
Time taken to sort 4 numbers is 0.000000 Secs
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 14500
3:To exit
Enter your choice:2

Time taken to sort 500 numbers is 0.000000 Secs
Time taken to sort 1500 numbers is 0.000000 Secs
Time taken to sort 2500 numbers is 0.000000 Secs
Time taken to sort 3500 numbers is 0.000000 Secs
Time taken to sort 4500 numbers is 0.015000 Secs
Time taken to sort 5500 numbers is 0.016000 Secs
Time taken to sort 6500 numbers is 0.000000 Secs
Time taken to sort 7500 numbers is 0.016000 Secs
Time taken to sort 8500 numbers is 0.016000 Secs
Time taken to sort 9500 numbers is 0.031000 Secs
Time taken to sort 10500 numbers is 0.047000 Secs
Time taken to sort 11500 numbers is 0.031000 Secs
Time taken to sort 12500 numbers is 0.063000 Secs
Time taken to sort 13500 numbers is 0.048000 Secs
Time taken to sort 14500 numbers is 0.079000 Secs
1:For manual entry of N value and array elements
2:To display time taken for sorting number of elements N in the range 500 to 14500
3:To exit
Enter your choice:|
```

Graph Output:-

