Aim: Generate a large number of elements randomly and sort all the elements in ascending order using Selection Sort. Analyze the time complexity for best, average and worst case

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

void insertionSort(int a[], int length){
    int i, index, p, j, temp;
    for(i=1; i<length; i++){
        p = a[i];
        for(j=i-1; (p < a[j] && j >= 0 ); j--){
            a[j+1]=a[j];
            a[j] = p;
        }
    }
}

void main(){

    int n,i;
    int *array;
    int t1,t2;
    float t3,t4;
    int tt1,tt2;
    struct timeval tv;
    struct timezone tz;

    printf("\nHow many values You want to enter:");
    scanf("%d",&n);

    array = (int*)malloc(n*sizeof(int));

    for(i=0 ; i < n ;i++){
        //array[i]=rand()%1000;        //For Average Case
        array[i]=i+1;              //For Best Case
        //array[i]=n-i;             //For Worst Case
    }

    gettimeofday(&tv,&tz);
    tt1 = tv.tv_sec;
    t1 = tv.tv_usec;
```

```
    insertionSort(array,n);

    gettimeofday(&tv,&tz);
    tt2 = tv.tv_sec;
    t2 = tv.tv_usec;

    tt1 = tt1%100;
    t3 = (float)t1*(0.000001);
    t3 = t3 + (float)tt1;
    tt2 = tt2%100;
    t4 = (float)t2*(0.000001);
    t4 = t4 + (float)tt2;

    printf("\nStart : %d.%d",tt1,t1);
    printf("\nEnd   : %d.%d",tt2,t2);

    if(t4>t3)
        printf("\nDifferenc1 : %f",t4-t3);
    else
        printf("\nDifferenc2 :  %f",t3-t4);

    free(array);
}
```

Time Analysis :

Table :

Total Elements	Best Case	Average Case	Worst Case
5000	0.007152	0.029984	0.061526
10000	0.008763	0.099921	0.247830
15000	0.009937	0.239868	0.463412
25000	0.011402	0.693562	1.079605

Graph :