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 swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int arr[], int n) {
    int i, j, min;

    for (i = 0; i < n-1; i++){
        min = i;
        for (j = i+1; j < n; j++){
            if (arr[j] < arr[min])
                min = j;
        }
        swap(&arr[min], &arr[i]);
    }
}

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

void main(){

    int n,i;
    int *array;
```

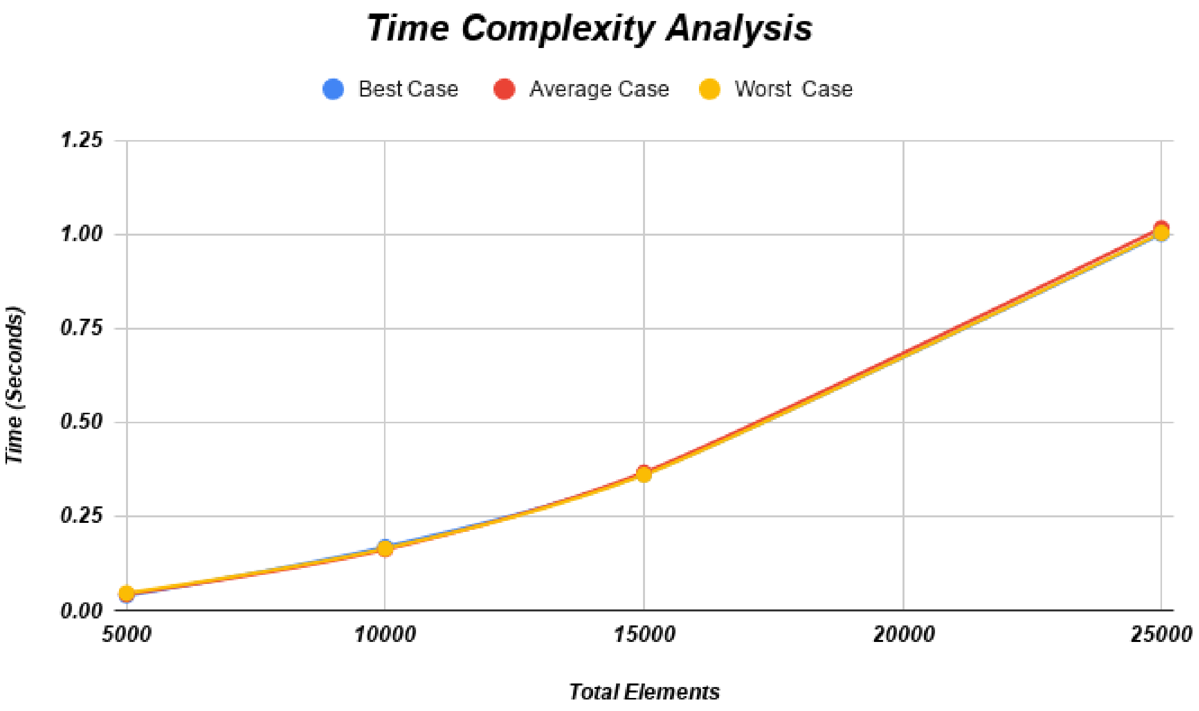
```c
    int t1,t2;
    float t3,t4;
    int tt1,tt2;
    struct timeval tv;
    struct timezone tz;

    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;

    selectionSort(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	0.040246	0.043549	0.046087
10000	0.168904	0.162047	0.164611
15000	0.361891	0.366562	0.360412
25000	1.000772	1.016258	1.003245

Graph :


Time Complexity Analysis
● Best Case
● Average Case
● Worst Case
Time (Seconds)
Total Elements