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

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


void bubbleSort(int b[],int n){
    int temp,i,j,flag;
    for(i = 0; i < n-1 ; i++){
        flag=0;
        for(j = 0 ; j < n-1-i ; j++){
            if(b[j]>b[j+1]){
                temp=b[j];
                b[j]=b[j+1];
                b[j+1]=temp;
                flag=1;
            }
        }
        if(flag==0)
            break;
    }
}

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;
    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;

    bubbleSort(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	0.001001	0.072643	0.081348
10000	0.006468	0.336206	0.349962
15000	0.005184	0.711798	0.669435
25000	0.013985	2.108708	2.494321

Graph :