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

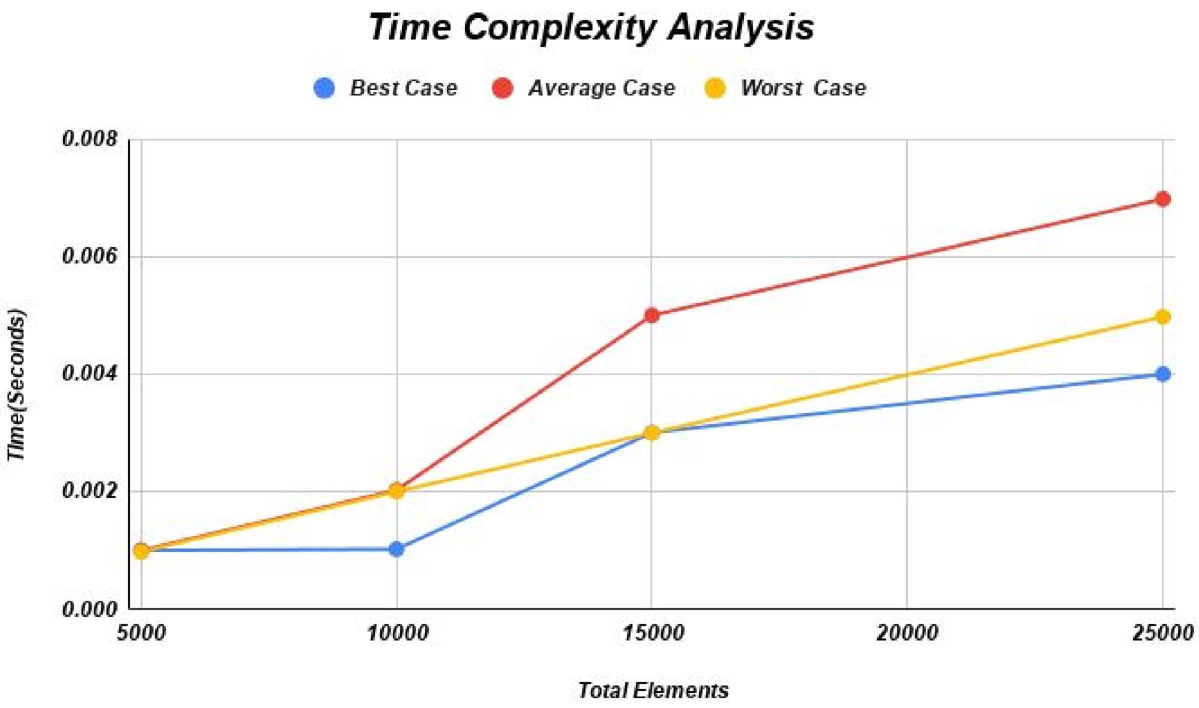
```c
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

void merge(int arr[], int l, int m, int r){
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    i = 0;
    j = 0;
    k = l;
    while (i < n1 && j < n2){
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else{
            arr[k] = R[j];
            j++;
        }
        k++;
    }
    while (i < n1){
        arr[k] = L[i];
        i++;
        k++;
    }
    while (j < n2){
        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int lef, int rig){
    if (lef < rig) {
        int mid = lef + (rig - lef) / 2;
        mergeSort(arr, lef, mid);
        mergeSort(arr, mid + 1, rig);
        merge(arr, lef, mid, rig);
    }
}

void shuffle(int *array, size_t n){
    if (n > 1) {
        size_t i;
        for (i = 0; i < n - 1; i++) {
          size_t j = i + rand() / (RAND_MAX / (n - i) + 1);
          int t = array[j];
          array[j] = array[i];
          array[i] = t;
        }
    }
}



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]=i+1;          //For Best Case
        array[i]=n-i;            //For Worst Case
    }
    //shuffle(array,n);          //For Average Case

    gettimeofday(&tv,&tz);
    tt1 = tv.tv_sec;
    t1 = tv.tv_usec;
    printf("\nStart : %d.%d",tt1,t1);
    mergeSort(array,0,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("\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	0.000999	0.000997	0.000977
10000	0.001019	0.002023	0.001998
15000	0.003003	0.004999	0.002998
25000	0.003998	0.006979	0.004975

Graph :


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