

## MERGE SORT:

```
#include<bits/stdc++.h>
using namespace std;
const int N=1e5+1;
int a[N];
int Noc=0;
void merge(int l,int r,int mid)
{
    int l_size=mid-l+1;
    int left_arr[l_size+1];
    int r_size=r-mid;
    int right_arr[r_size+1];

    for(int i=0;i<l_size;i++)
    {
        left_arr[i]=a[i+l];
    }
    for(int i=0;i<r_size;i++)
    {
        right_arr[i]=a[i+mid+1];
    }
    left_arr[l_size]=right_arr[r_size]=INT_MAX;

    int l_ind=0;
    int r_ind=0;
    for(int i=l;i<=r;i++)
    {
        if(left_arr[l_ind]<=right_arr[r_ind])
        {
            a[i]=left_arr[l_ind];
            l_ind++;
        }
        else
        {
            a[i]=right_arr[r_ind];
            r_ind++;
        }
        Noc++;
    }
}

void mergeSort(int l,int r)
{
    if(l==r) return;
    int mid=(l+r)/2;
    mergeSort(l,mid);
    mergeSort(mid+1,r);
}
```

```

merge(l,r,mid);
}
int main()
{
int t=100;
srand(time(0));
ofstream outdata;
ofstream outdata2;
outdata.open("Mergesort.txt");
outdata2.open("MergesortComparison.txt");
outdata2<<"ITERATION NO. |"<<"\t| I/P size\t"<<"\t| Comparisons\t"<<"\t| Time
(sec)"<<endl<<endl;
while(t--)
{
clock_t start, end;
cout<<endl;
outdata<<endl;
outdata<<"\n
n-----\n\
n";
cout<<"\tITERATION NO.-->"<<100-t<<"\n";
outdata<<"\tITERATION NO.-->"<<100-t<<"\n";

start = clock();
int lb = 20, ub = 1000;
int l=30,u=1000;
int n=(rand() % (u - l + 1)) + 1 ;
cout<<endl;
cout<<"\tSize of array-->"<<n<<"\n\n";
outdata<<endl;
outdata<<"\tSize of array-->"<<n<<"\n\n";
for(int i=0;i<n;i++)
{
int x=(rand() % (ub - lb + 1)) + lb;
cout<<x<<" ";
outdata<<x<<" ";
a[i]=x;
}
cout<<endl;
outdata<<endl;
mergeSort(0,n-1);
cout<<"\n\tSorted array\n\n";
outdata<<"\n\nS O R T E D A R R A Y\n\n";
for(int i=0;i<n;i++)
{
cout<<a[i]<<" , ";
outdata<<a[i]<<" , ";
}
}

```

```

cout<<"\n\nTOTAL NUMBER OF COMPARISONS--> "<<Noc<<"\n\n";
outdata<<"\n\nTOTAL NUMBER OF COMPARISONS--> "<<Noc<<"\n\n";
end = clock();
double time_taken = double(end - start) / double(CLOCKS_PER_SEC);
cout << "Time taken by ITERATION "<<100-t<<" is : " << fixed
<< time_taken << setprecision(5);
cout << " sec " << endl<<endl;

outdata << "Time taken by ITERATION "<<100-t<<" is : " << fixed
<< time_taken << setprecision(5);
outdata << " sec " << endl<<endl;

outdata2<<"\t"<<100-t<<"\t"<<n<<"\t"<<Noc<<"\t"<< fixed<<time_taken <<
setprecision(5)<<"\n";
Noc=0;
cout<<"\n
n-----\n\n";
outdata<<"\n-----\n\
n";
}
outdata.close();
}

```

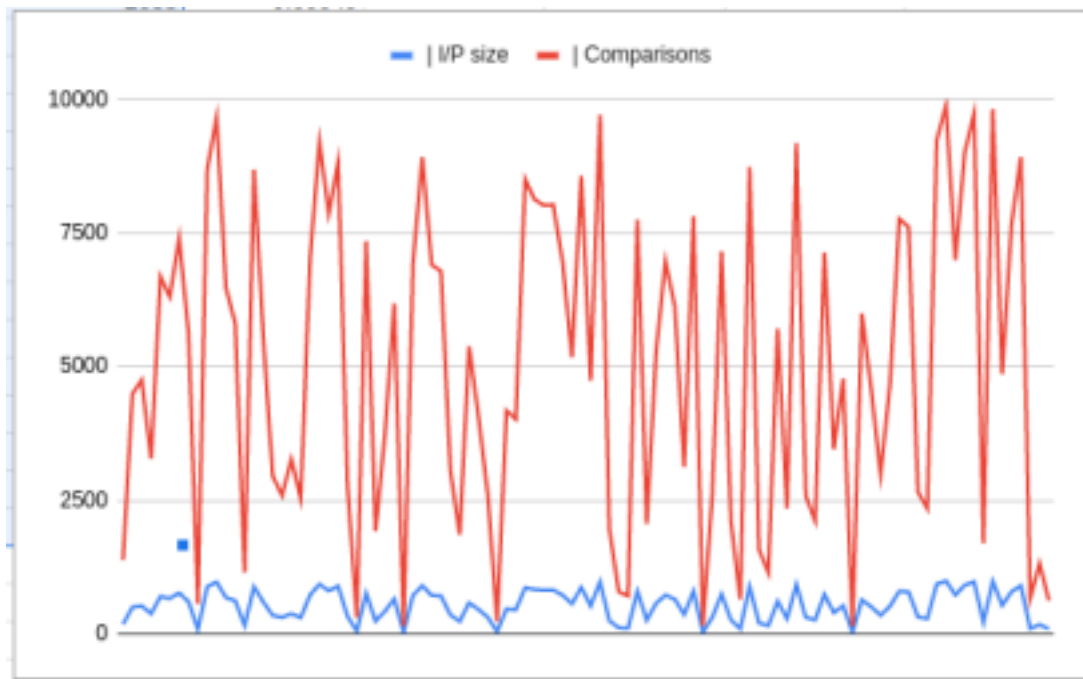
## OUTPUT :

```

1 |
2 |
3 |
4 | -----
5 | ITERATION NO.-->1
6 | Size of array-->100
7 |
8 | 819 962 657 64 125 193 386 343 389 33 679 371 238 799 377 903 816 224 553 626 682 840 385 45 483 800 408 172 878 623 951 679 525 475 715 838 640 998 961 37 30 848 389 148 546 654 938 381 858 49
9 |
10 | SORTED ARRAY
11 |
12 | 30 , 32 , 33 , 37 , 43 , 44 , 45 , 50 , 50 , 53 , 53 , 53 , 60 , 61 , 64 , 69 , 71 , 80 , 87 , 89 , 94 , 95 , 96 , 118 , 127 , 132 , 146 , 151 , 154 , 167 , 172 , 182 , 189 , 195 , 195 , 202 ,
13 |
14 | TOTAL NUMBER OF COMPARISONS--> 1382
15 |
16 | Time taken by ITERATION 1 is : 0.000123 sec
17 |
18 | -----
19 |
20 |
21 |
22 | -----
23 | ITERATION NO.-->2
24 | Size of array-->500
25 |
26 | 254 862 297 871 298 836 378 540 757 410 347 618 965 338 858 24 181 578 587 368 337 51 687 396 536 879 296 485 343 885 21 576 834 299 446 812 823 712 943 487 144 187 954 957 413 768 28 483 337 4
27 |
28 | SORTED ARRAY
29 |
30 | 20 , 21 , 21 , 24 , 29 , 30 , 33 , 33 , 37 , 39 , 39 , 40 , 43 , 43 , 47 , 51 , 54 , 55 , 57 , 58 , 59 , 61 , 62 , 63 , 67 , 71 , 74 , 74 , 75 , 75 , 79 , 79 , 80 , 81 , 82 , 84 , 85 , 87 , 88
31 |
32 | TOTAL NUMBER OF COMPARISONS--> 6480
33 |
34 | Time taken by ITERATION 2 is : 0.00036 sec
35 |
36 | -----
37 |
38 |
39 |
40 | -----
41 | ITERATION NO.-->3
42 | Size of array-->525
43 |
44 | 312 966 968 87 410 542 384 957 586 583 682 785 498 787 63 79 680 315 878 358 388 781 159 27 788 573 86 881 689 86 881 838 81 252 887 881 736 381 757 387 783 326 881 167 831 818 317 682 147 87 8
45 |
46 | SORTED ARRAY
47 |

```

## GRAPH :



## INSERTION SORT :

```
#include <bits/stdc++.h>
using namespace std;
int a[1000];
int Noc=0;

void insertionSort(int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = a[i];
        j = i - 1;
        while (j >= 0 && a[j] > key)
        {
            a[j + 1] = a[j];
            j = j - 1;
            Noc++;
        }
        a[j + 1] = key;
    }
}

int main()
{
    int t=100;
```

```

srand(time(0));
ofstream outdata;
ofstream outdata2;
outdata.open("Insertionsort.txt");
outdata2.open("InsertionSortComparison.txt");
outdata2<<"ITERATION NO. |"<<"\t| I/P size\t"<<"\t| Comparisons\t"<<"\t| Time
(sec)"<<endl<<endl;
while(t-->0)
{
clock_t start, end;
cout<<endl;
outdata<<endl;
outdata<<"\n
n-----\n\n";
cout<<"\tITERATION NO.-->"<<100-t<<"\n";
outdata<<"\tITERATION NO.-->"<<100-t<<"\n";
start = clock();
int lb = 0, ub = 5000;
int l=30,u=1000;
int n=(rand() % (u - l + 1)) + l ;
cout<<endl;
cout<<"\tSize of array-->"<<n<<"\n\n";
outdata<<endl;
outdata<<"\tSize of array-->"<<n<<"\n\n";
for(int i=0;i<n;i++)
{
int x=(rand() % (ub - lb + 1)) + lb;
cout<<x<<" ";
outdata<<x<<" ";
a[i]=x;
}
cout<<endl;
outdata<<endl;
insertionSort(n);
cout<<"\n\tSorted array\n\n";
outdata<<"\n\nS O R T E D A R R A Y\n\n";
for (int i = 0; i < n; i++)
{
cout << a[i] << " , ";
outdata << a[i] << " , ";
}
cout << endl;
cout<<"\n\nTOTAL NUMBER OF COMPARISONS--> " <<Noc<<"\n\n";

outdata<<"\n\nTOTAL NUMBER OF COMPARISONS--> " <<Noc<<"\n\n";
end = clock();
double time_taken = double(end - start) / double(CLOCKS_PER_SEC);
cout << "Time taken by ITERATION "<<100-t<<" is : " << fixed

```

```

<< time_taken << setprecision(5);
cout << " sec " << endl<<endl;

outdata << "Time taken by ITERATION "<<100-t<<" is : " << fixed
<< time_taken << setprecision(5);
outdata << " sec " << endl<<endl;
outdata2<<"\t"<<100-t<<"\t"<<n<<"\t"<<Noc<<"\t"<< fixed<<time_taken <<
setprecision(5)<<"\n";
Noc=0;
cout<<"\n-----\n\
n";
outdata<<"\
n-----\n\n"; }
outdata.close();
}

```

## OUTPUT :

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

-----
ITERATION NO.-->1
Size of array-->400
2086 2073 4841 3798 4798 1927 3689 437 136 4318 4396 1387 4811 398 2518 848 3757 173 547 628 8888 3820 715 848 1888 4573 3833 3838 3887 4243 3867 3856 3314 3787 3637 1876 396 3786 3383 533 2815
SORTED ARRAY
26 , 39 , 43 , 71 , 74 , 79 , 83 , 126 , 123 , 128 , 128 , 158 , 178 , 200 , 233 , 235 , 245 , 262 , 288 , 298 , 292 , 299 , 299 , 327 , 338 , 348 , 358 , 358 , 368 , 368 , 374 , 375
TOTAL NUMBER OF COMPARISONS--> 33294
Time taken by ITERATION 1 is : 0.001351 sec
-----

ITERATION NO.-->2
Size of array-->375
3839 278 3833 333 3982 3835 3875 3823 326 33 2575 3888 4736 3583 4785 2881 3875 4758 1583 733 173 3823 2488 1738 538 3888 4548 88 1833 3386 1158 236 4823 4848 457 4887 3578 84 1888 4887 385 488
SORTED ARRAY
1 , 8 , 11 , 32 , 68 , 84 , 185 , 388 , 338 , 368 , 388 , 372 , 183 , 388 , 235 , 223 , 238 , 234 , 237 , 278 , 294 , 382 , 384 , 323 , 328 , 329 , 338 , 388 , 383 , 386 , 426 , 453 , 458 , 457
TOTAL NUMBER OF COMPARISONS--> 35430
Time taken by ITERATION 2 is : 0.00084 sec
-----

ITERATION NO.-->3
Size of array-->381
872 3343 3959 2947 173 638 336 3473 3387 88 982 4887 3888 438 4875 3844 2328 4879 4731 4838 1888 2484 1871 4887 1888 3838 2838 2782 4383 4558 51 275 3581 4774 3885 4487 381 4338 1888 3778 188 3
SORTED ARRAY

```

## GRAPH :

