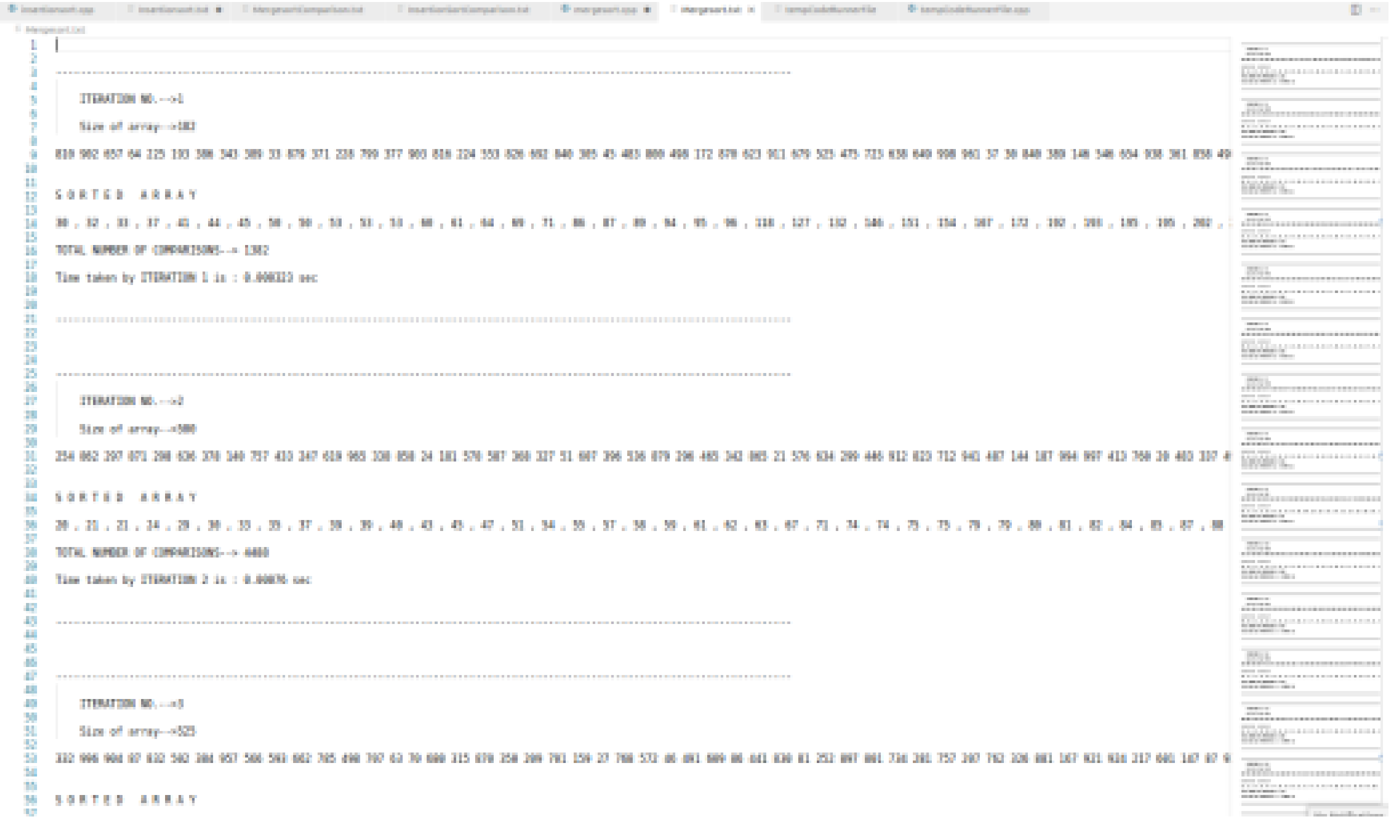
Khushi Chhatwani CSC/21/55

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);
```

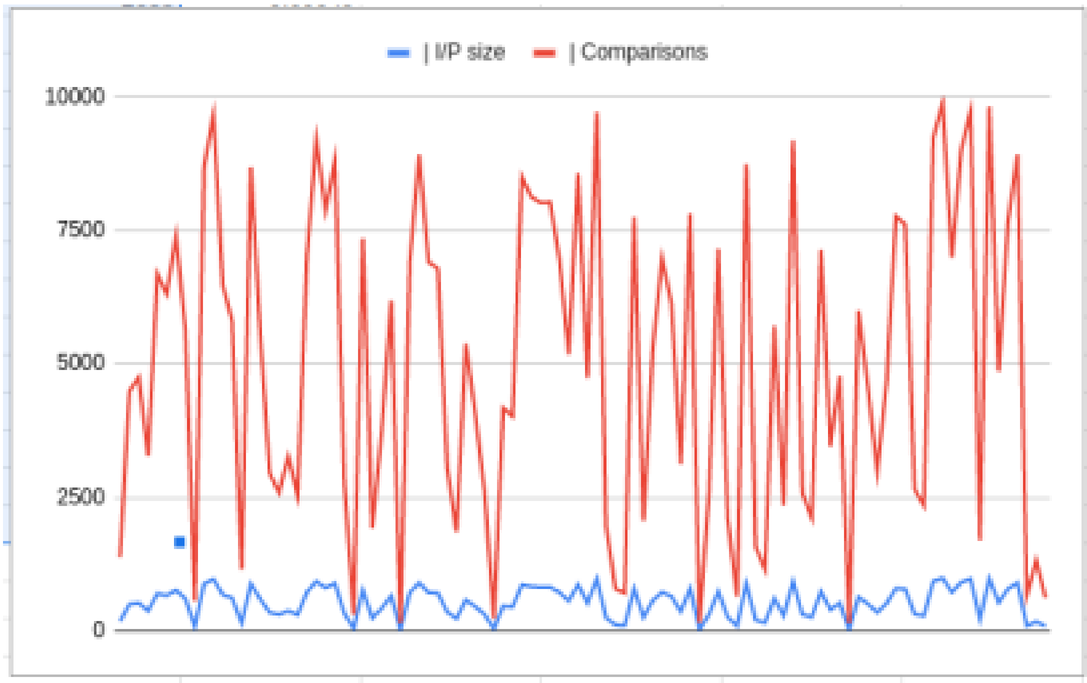
```
merge(l,r,mid);
 }
 int main()
 {
 int t=100;
 srand(time(0));
 ofstream outdata;
 ofstream outdata2;
 outdata.open("Mergesort.txt");
 outdata2.open("MergesortComparison.txt");
 outdata2<<"ITERATION NO. |"<<"\t| I/P size\t"<<"\t| Comparisons\t"<<"\t| Time
 (sec)"<<endl<<endl;
 while(t--)
 {
 clock_t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

OUTPUT :



```
-----------------------------------------------------------------------------------------

    ITERATION NO.--->1

    Size of array--->182

[illegible]

S O R T E D   A R R A Y

[illegible]

TOTAL NUMBER OF COMPARISONS--> 1382

Time taken by ITERATION 1 is : 0.000323 sec

-----------------------------------------------------------------------------------------



-----------------------------------------------------------------------------------------

    ITERATION NO.--->2

    Size of array--->500

[illegible]

S O R T E D   A R R A Y

[illegible]

TOTAL NUMBER OF COMPARISONS--> 4460

Time taken by ITERATION 2 is : 0.00076 sec

-----------------------------------------------------------------------------------------



-----------------------------------------------------------------------------------------

    ITERATION NO.--->3

    Size of array--->525

[illegible]

S O R T E D   A R R A Y
```

GRAPH :

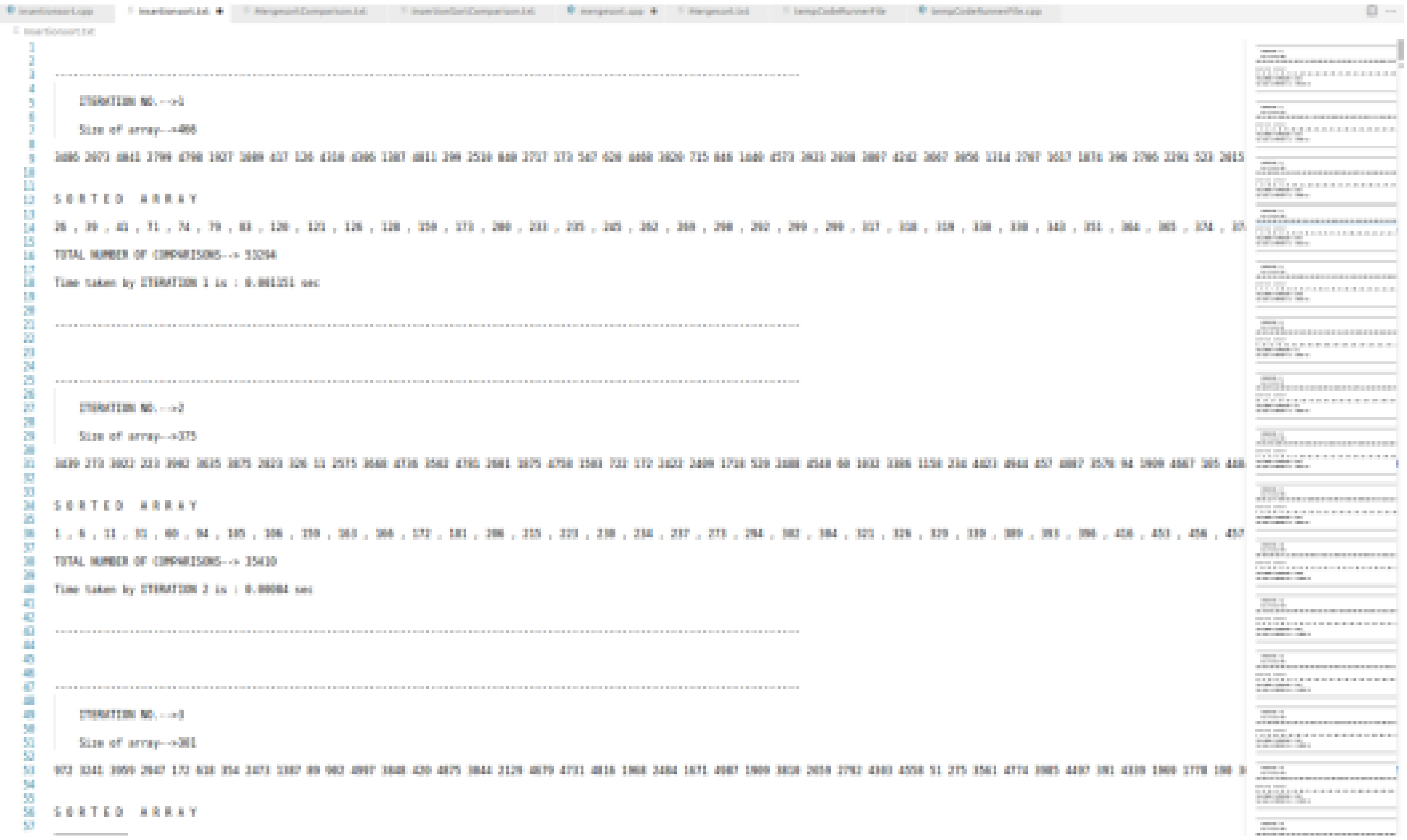


INSERTION SORT :

```
#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;
```

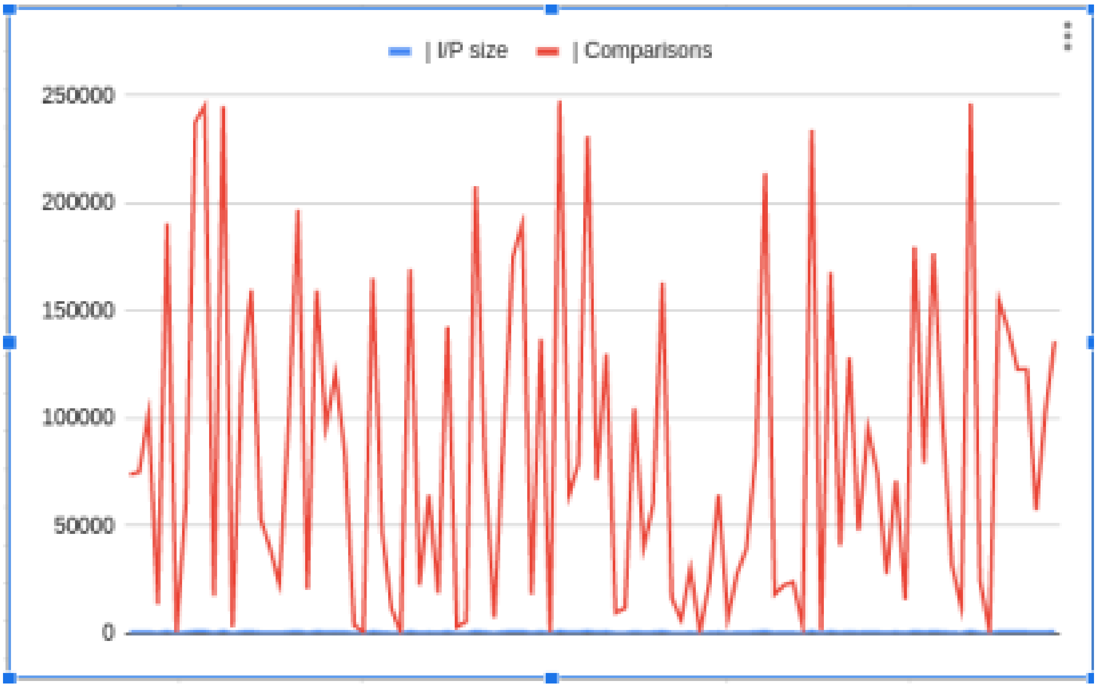
```
srand(time(0));
ofstream outdata;
ofstream outdata2;
outdata.open("Insertionsort.txt");
outdata2.open("InsertionSortComparison.txt");
outdata2<<"ITERATION NO. |"<<"\t| I/P size\t"<<"\t| Comparisons\t"<<"\t| Time
 (sec)"<<endl<<endl;
while(t--)
 {
clock_t start, end;
cout<<endl;
outdata<<endl;
outdata<<"\
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 :



```


---------------------------------------------------------------------------------------------------

    ITERATION NO.--->1

    Size of array--->400

3085 3973 4841 3799 4798 1927 3809 437 126 4318 4306 1387 4811 299 2510 840 2717 373 547 620 4468 3026 715 846 1660 4573 2023 2038 2867 4242 3667 3856 1314 2707 3617 1874 396 2705 2291 523 2615


S O R T E D   A R R A Y

26 , 39 , 41 , 71 , 74 , 79 , 83 , 120 , 121 , 126 , 128 , 150 , 173 , 200 , 231 , 235 , 245 , 262 , 269 , 290 , 292 , 299 , 299 , 317 , 318 , 319 , 330 , 338 , 343 , 351 , 364 , 365 , 374 , 37

TOTAL NUMBER OF COMPARISONS--> 53294

Time taken by ITERATION 1 is : 0.001151 sec


---------------------------------------------------------------------------------------------------




---------------------------------------------------------------------------------------------------

    ITERATION NO.--->2

    Size of array--->275

3439 273 3022 223 3982 3635 3075 2823 326 11 2575 3640 4736 3502 4781 2661 3075 4758 1503 732 172 3422 2409 1718 520 3488 4548 60 1032 3386 1150 234 4423 4944 457 4887 3570 94 3909 4667 305 448


S O R T E D   A R R A Y

1 , 6 , 11 , 31 , 60 , 94 , 105 , 106 , 159 , 163 , 166 , 172 , 181 , 206 , 215 , 223 , 230 , 234 , 237 , 271 , 294 , 302 , 304 , 321 , 326 , 329 , 339 , 389 , 393 , 396 , 416 , 451 , 456 , 457

TOTAL NUMBER OF COMPARISONS--> 35410

Time taken by ITERATION 2 is : 0.00084 sec


---------------------------------------------------------------------------------------------------




---------------------------------------------------------------------------------------------------

    ITERATION NO.--->3

    Size of array--->301

972 3241 3959 2947 172 618 354 3471 3387 89 982 4997 3848 420 4875 3844 2129 4879 4731 4816 1968 2484 1671 4987 1909 3818 2019 2792 4343 4558 51 275 3561 4774 3985 4497 391 4339 1969 1778 198 3


S O R T E D   A R R A Y
```

GRAPH :


| I/P size
| Comparisons
250000
200000
150000
100000
50000
0