Khushi Chhatwani CSC/21/55

HEAP SORT :

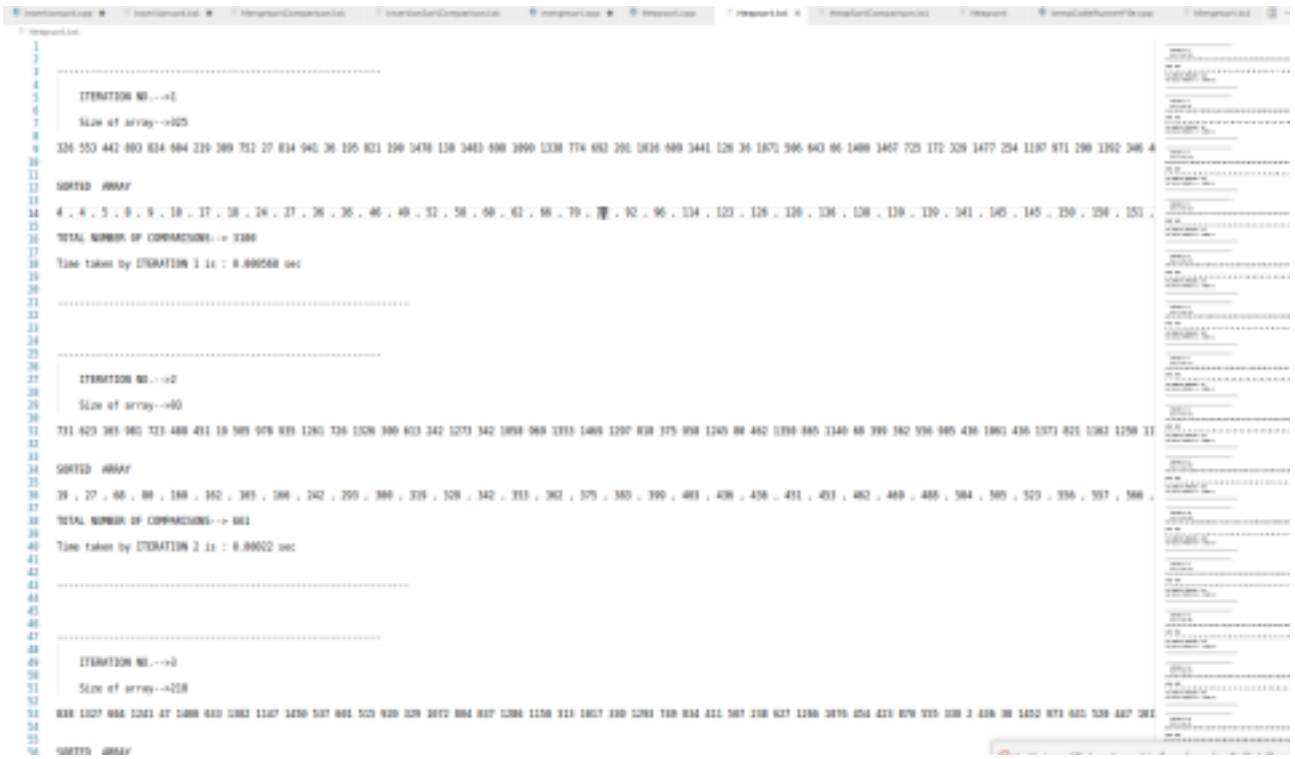
```
#include <bits/stdc++.h>
using namespace std;
const int N=1e5+1;
int arr[N];
int Noc = 0;
void heapify(int n, int i)
{
int lg = i;
int rc = 2 * lg + 2;
int lc = 2 * lg + 1;
if (rc < n && arr[rc] > arr[lg])
{
Noc++;
lg = rc;
}
if (lc < n && arr[lc] > arr[lg])
{
lg = lc;
Noc++;
}
if (lg != i)
{
swap(arr[lg], arr[i]);
heapify(n, lg);
}
}
void heapsort(int n)
{
int size = n;
while (size > 0)
{
size--;
swap(arr[size], arr[0]);
heapify(size, 0);
}
}
int main()
{
int t = 100;
srand(time(0));
ofstream outdata;
ofstream outdata2;
outdata.open("Heapsort.txt");
outdata2.open("HeapSortComparison.txt");
```

GARIMA ROHILLA CSC/21/37

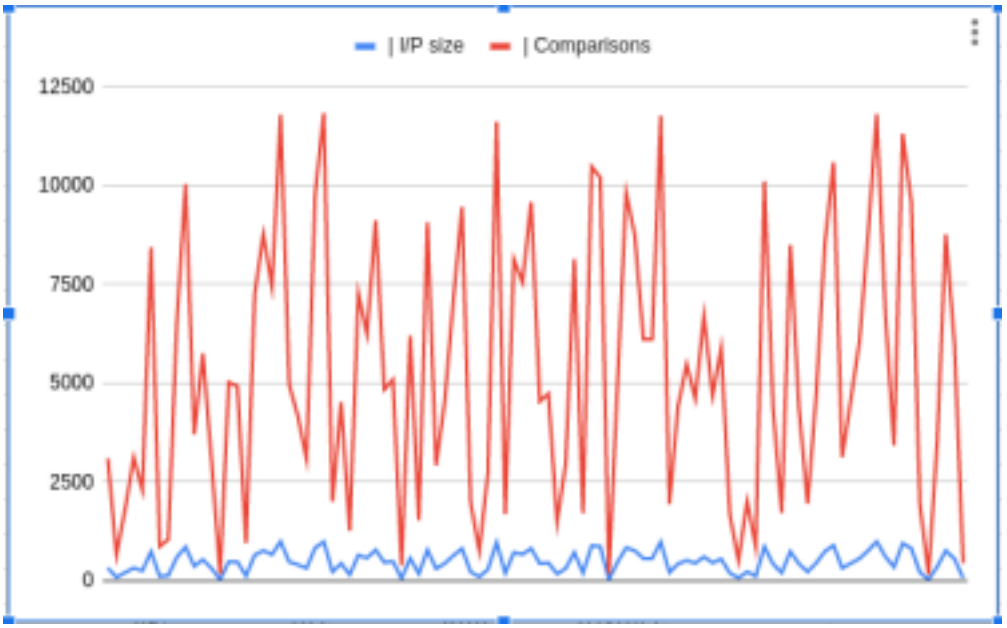
```
outdata2<<"ITERATION NO. |"<<"\t| I/P size\t"<<"\t| Comparisons\t"<<"\t| Time
(sec)"<<endl<<endl;

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 = 1500;
int l = 30, u = 1000;
int k = (rand() % (u - l + 1)) + l;
cout << endl;
cout << "\tSize of array-->" << k<< "\n\n";
outdata << endl;
outdata << "\tSize of array-->" << k << "\n\n";
for (int i = 0; i < k; i++)
{
int x = (rand() % (ub - lb + 1)) + lb;
cout << x << " ";
outdata << x << " ";
arr[i] = x;
}
cout << endl;
outdata << endl;
for (int i = k / 2 - 1; i >= 0; i--)
{
heapify(k, i);
}
heapsort(k);
cout << "\n\tSorted array\n\n";
outdata << "\n\nSORTED ARRAY\n\n";
for (int i = 0; i < k; i++)
{
cout << arr[i] << " ";
outdata << arr[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"<<k<<"\t"<<Noc<<"\t"<< fixed<<time_taken <<
setprecision(5)<<"\n";

Noc = 0;
cout << "\n-----------------------------------------------------------\n\n";
outdata << "\n-----------------------------------------------------------------\n\n";
}
outdata.close();
}
```

OUTPUT :



GRAPH :


| I/P size
| Comparisons
12500
10000
7500
5000
2500
0