Practical 2

(Tanu Soni,88067)

Implement Heap Sort (The program should report the number of comparisons) test runs the algorithm on 100 different inputs of sizes varying from 30 to 1000.Count the number of comparisons and draw the graph. Compare it with a graph of nlogn.

Code

```
#include <bits/stdc++.h>
using namespace std;
int cnt = 0;

void heapify(int arr[], int n, int i)
{
        cnt++;
        int largest = i;
        int left = 2 * i + 1;
        int right = 2 * i + 2;

        if (left < n && arr[left] > arr[largest])
                largest = left;

        if (right < n && arr[right] > arr[largest])
                largest = right;

        if (largest != i)
        {
                swap(arr[i], arr[largest]);
                heapify(arr, n, largest);
        }
}

void heapsort(int arr[], int n)
{
        // Building max-heap
        for (int i = n / 2 - 1; i >= 0; i--)
                heapify(arr, n, i);

        // heap sort
```

```
        for (int i = n - 1; i >= 0; i--)
        {
                swap(arr[0], arr[i]);

                // heapify root element
                heapify(arr, i, 0);
        }
}

int main()
{
        int size;
        ofstream fout("MyExcel.csv");
        fout << "Size"
             << ","
             << "Comparisons" << endl;
        srand(time(0));
        for (int i = 0; i < 100; i++)
        {
                size = rand() % 971 + 30;
                int Array[size] = {0};
                for (int j = 0; j < size; j++)
                {
                        Array[j] = rand() % 10000;
                }
                heapsort(Array, size);
                fout << size << "," << cnt << endl;
                cnt = 0;

        }

        return 0;
}
```

Output

```
Size,Comparisons
609,5405
836,7795
728,6675
664,5998
```

```
710,6465
934,8880
692,6278
558,4888
204,1486
90,558
519,4455
556,4863
496,4276
349,2796
985,9424
327,2633
35,167
709,6455
464,3930
710,6438
330,2637
231,1717
779,7212
369,3006
503,4313
535,4631
286,2198
986,9483
848,7937
758,7016
36,176
923,8735
813,7569
711,6494
750,6875
781,7235
536,4620
357,2922
180,1296
578,5080
137,915
703,6388
747,6857
595,5271
513,4430
600,5338
```

```
803,7425
451,3812
608,5386
371,3030
520,4528
547,4779
254,1938
87,532
907,8547
437,3687
41,198
83,516
622,5583
118,774
615,5461
828,7725
453,3856
636,5699
634,5693
116,747
860,8087
510,4392
906,8564
375,3077
802,7481
126,833
392,3236
154,1063
576,5046
860,8075
719,6576
786,7284
952,9079
53,292
888,8365
624,5566
755,6947
255,1938
635,5706
834,7805
145,1004
791,7322
```

```
202,1484
869,8185
735,6729
871,8205
167,1181
675,6110
36,183
385,3204
766,7080
946,9002
412,3436
296,2346
```

Graph