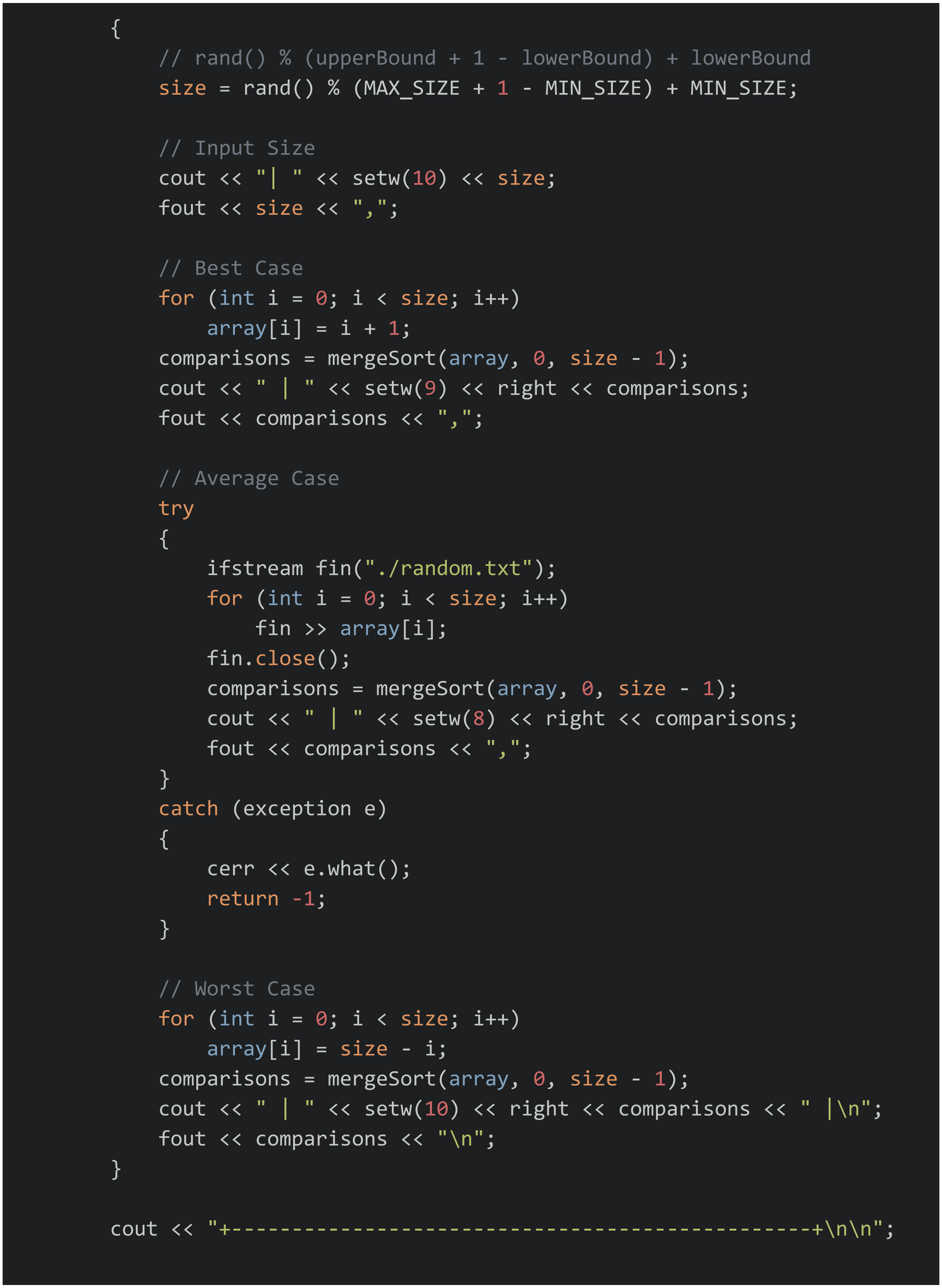
Practical 1A

(Tanu Soni,88067)

Implement Merge 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

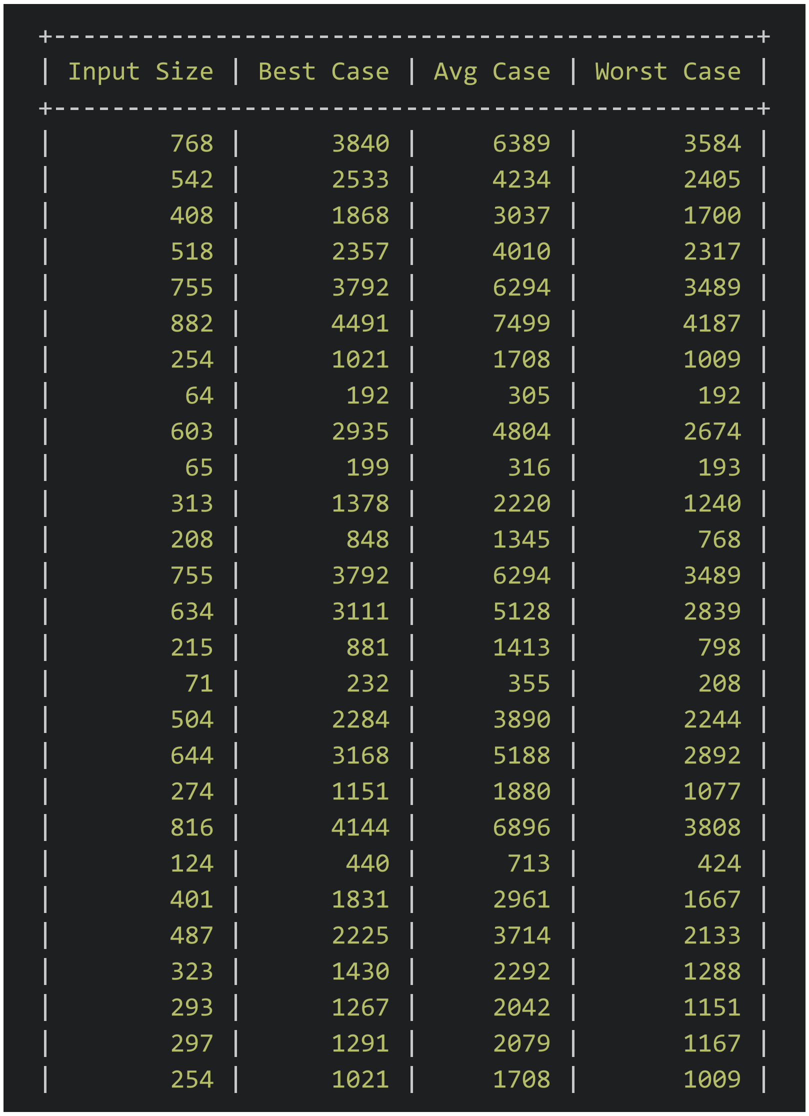
```
#include <cstdlib>
#include <fstream>
#include <iomanip>
#include <iostream>

#define MIN_SIZE 30
#define MAX_SIZE 1000

using namespace std;

int mergeSort(int *, int, int);
int merge(int *, int, int, int);

int main()
{
    try
    {
        srand(time(0));

        int array[MAX_SIZE];
        int size, comparisons;

        ofstream fout("./results.csv");

        cout << "+------------------------------------------------+\n";
        cout << "| Input Size | Best Case | Avg Case | Worst Case |\n";
        cout << "+------------------------------------------------+\n";

        fout << "size,best,avg,worst\n";

        for (int i = 0; i < 100; i++)
```



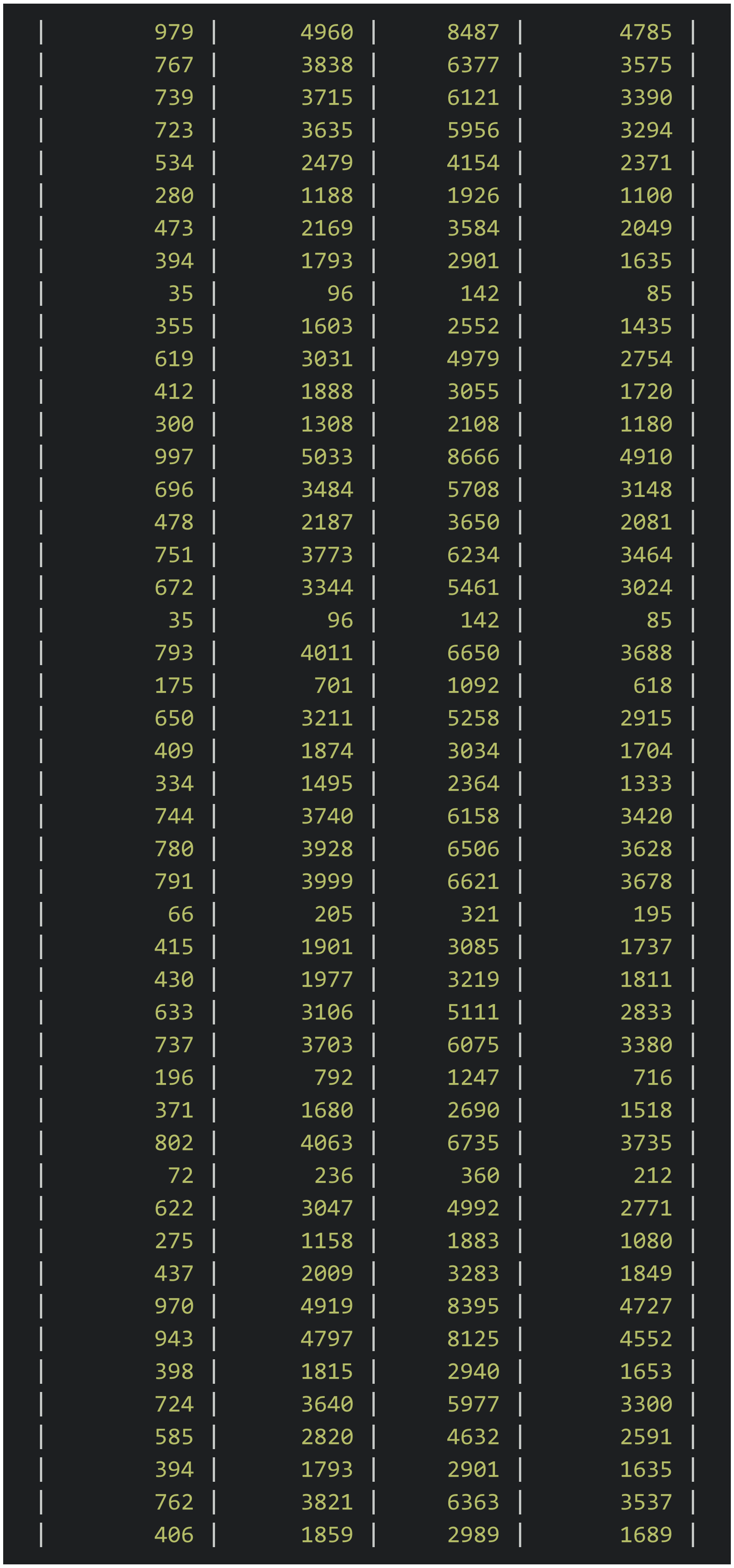
```
    {
        // rand() % (upperBound + 1 - lowerBound) + lowerBound
        size = rand() % (MAX_SIZE + 1 - MIN_SIZE) + MIN_SIZE;

        // Input Size
        cout << "| " << setw(10) << size;
        fout << size << ",";

        // Best Case
        for (int i = 0; i < size; i++)
            array[i] = i + 1;
        comparisons = mergeSort(array, 0, size - 1);
        cout << " | " << setw(9) << right << comparisons;
        fout << comparisons << ",";

        // Average Case
        try
        {
            ifstream fin("./random.txt");
            for (int i = 0; i < size; i++)
                fin >> array[i];
            fin.close();
            comparisons = mergeSort(array, 0, size - 1);
            cout << " | " << setw(8) << right << comparisons;
            fout << comparisons << ",";
        }
        catch (exception e)
        {
            cerr << e.what();
            return -1;
        }

        // Worst Case
        for (int i = 0; i < size; i++)
            array[i] = size - i;
        comparisons = mergeSort(array, 0, size - 1);
        cout << " | " << setw(10) << right << comparisons << " |\n";
        fout << comparisons << "\n";
    }

    cout << "+-------------------------------------------------+\n\n";
```

```
        fout.close();

        return 0;
    }
    catch (exception e)
    {
        cerr << e.what();
        return -1;
    }
}

int mergeSort(int *array, int beg, int end)
{
    int comparisons = 0;
    if (beg < end)
    {
        int mid = (beg + end) / 2;
        comparisons += mergeSort(array, beg, mid);
        comparisons += mergeSort(array, mid + 1, end);
        comparisons += merge(array, beg, mid, end);
    }
    return comparisons;
}

int merge(int *array, int beg, int mid, int end)
{
    int comparisons = 0;

    int n1 = mid - beg + 1;
    int n2 = end - mid;
    int L[n1 + 1], R[n2 + 1];

    for (int i = 0; i < n1; i++)
        L[i] = array[beg + i];
    for (int j = 0; j < n2; j++)
        R[j] = array[mid + 1 + j];

    L[n1] = R[n2] = INT16_MAX;

    for (int i = 0, j = 0, k = beg; k <= end; k++)
    {
        if (L[i] != INT16_MAX &&
            R[j] != INT16_MAX)
            comparisons++;

        if (L[i] <= R[j])
            array[k] = L[i++];
        else
            array[k] = R[j++];
    }

    return comparisons;
}
```

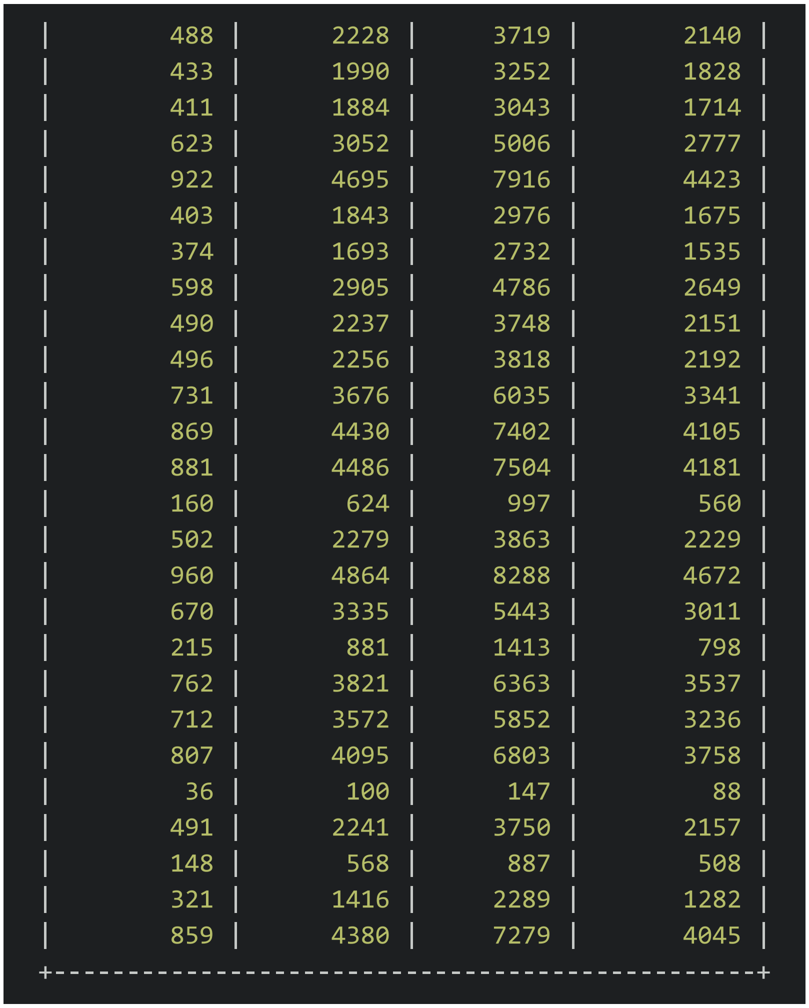
Output



```
+-----------------------------------------------+
| Input Size | Best Case | Avg Case | Worst Case |
+-----------------------------------------------+
|        768 |      3840 |     6389 |       3584 |
|        542 |      2533 |     4234 |       2405 |
|        408 |      1868 |     3037 |       1700 |
|        518 |      2357 |     4010 |       2317 |
|        755 |      3792 |     6294 |       3489 |
|        882 |      4491 |     7499 |       4187 |
|        254 |      1021 |     1708 |       1009 |
|         64 |       192 |      305 |        192 |
|        603 |      2935 |     4804 |       2674 |
|         65 |       199 |      316 |        193 |
|        313 |      1378 |     2220 |       1240 |
|        208 |       848 |     1345 |        768 |
|        755 |      3792 |     6294 |       3489 |
|        634 |      3111 |     5128 |       2839 |
|        215 |       881 |     1413 |        798 |
|         71 |       232 |      355 |        208 |
|        504 |      2284 |     3890 |       2244 |
|        644 |      3168 |     5188 |       2892 |
|        274 |      1151 |     1880 |       1077 |
|        816 |      4144 |     6896 |       3808 |
|        124 |       440 |      713 |        424 |
|        401 |      1831 |     2961 |       1667 |
|        487 |      2225 |     3714 |       2133 |
|        323 |      1430 |     2292 |       1288 |
|        293 |      1267 |     2042 |       1151 |
|        297 |      1291 |     2079 |       1167 |
|        254 |      1021 |     1708 |       1009 |
```

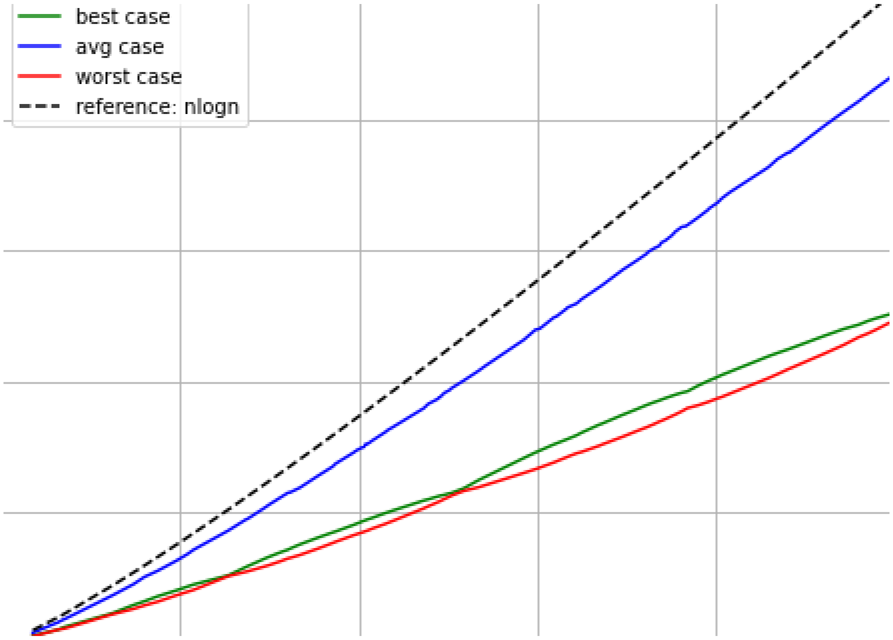


```
|         979 |       4960 |      8487 |        4785 |
|         767 |       3838 |      6377 |        3575 |
|         739 |       3715 |      6121 |        3390 |
|         723 |       3635 |      5956 |        3294 |
|         534 |       2479 |      4154 |        2371 |
|         280 |       1188 |      1926 |        1100 |
|         473 |       2169 |      3584 |        2049 |
|         394 |       1793 |      2901 |        1635 |
|          35 |         96 |       142 |          85 |
|         355 |       1603 |      2552 |        1435 |
|         619 |       3031 |      4979 |        2754 |
|         412 |       1888 |      3055 |        1720 |
|         300 |       1308 |      2108 |        1180 |
|         997 |       5033 |      8666 |        4910 |
|         696 |       3484 |      5708 |        3148 |
|         478 |       2187 |      3650 |        2081 |
|         751 |       3773 |      6234 |        3464 |
|         672 |       3344 |      5461 |        3024 |
|          35 |         96 |       142 |          85 |
|         793 |       4011 |      6650 |        3688 |
|         175 |        701 |      1092 |         618 |
|         650 |       3211 |      5258 |        2915 |
|         409 |       1874 |      3034 |        1704 |
|         334 |       1495 |      2364 |        1333 |
|         744 |       3740 |      6158 |        3420 |
|         780 |       3928 |      6506 |        3628 |
|         791 |       3999 |      6621 |        3678 |
|          66 |        205 |       321 |         195 |
|         415 |       1901 |      3085 |        1737 |
|         430 |       1977 |      3219 |        1811 |
|         633 |       3106 |      5111 |        2833 |
|         737 |       3703 |      6075 |        3380 |
|         196 |        792 |      1247 |         716 |
|         371 |       1680 |      2690 |        1518 |
|         802 |       4063 |      6735 |        3735 |
|          72 |        236 |       360 |         212 |
|         622 |       3047 |      4992 |        2771 |
|         275 |       1158 |      1883 |        1080 |
|         437 |       2009 |      3283 |        1849 |
|         970 |       4919 |      8395 |        4727 |
|         943 |       4797 |      8125 |        4552 |
|         398 |       1815 |      2940 |        1653 |
|         724 |       3640 |      5977 |        3300 |
|         585 |       2820 |      4632 |        2591 |
|         394 |       1793 |      2901 |        1635 |
|         762 |       3821 |      6363 |        3537 |
|         406 |       1859 |      2989 |        1689 |
```



```
|        488 |       2228 |      3719 |        2140 |
|        433 |       1990 |      3252 |        1828 |
|        411 |       1884 |      3043 |        1714 |
|        623 |       3052 |      5006 |        2777 |
|        922 |       4695 |      7916 |        4423 |
|        403 |       1843 |      2976 |        1675 |
|        374 |       1693 |      2732 |        1535 |
|        598 |       2905 |      4786 |        2649 |
|        490 |       2237 |      3748 |        2151 |
|        496 |       2256 |      3818 |        2192 |
|        731 |       3676 |      6035 |        3341 |
|        869 |       4430 |      7402 |        4105 |
|        881 |       4486 |      7504 |        4181 |
|        160 |        624 |       997 |         560 |
|        502 |       2279 |      3863 |        2229 |
|        960 |       4864 |      8288 |        4672 |
|        670 |       3335 |      5443 |        3011 |
|        215 |        881 |      1413 |         798 |
|        762 |       3821 |      6363 |        3537 |
|        712 |       3572 |      5852 |        3236 |
|        807 |       4095 |      6803 |        3758 |
|         36 |        100 |       147 |          88 |
|        491 |       2241 |      3750 |        2157 |
|        148 |        568 |       887 |         508 |
|        321 |       1416 |      2289 |        1282 |
|        859 |       4380 |      7279 |        4045 |
+---------------------------------------------------+
```

Graph


best case
avg case
worst case
reference: nlogn