Closest Pair of Points on 2-D Plane

Maciej Medyk

Z15197421

Project description

The Closest Pair of Points algorithm is designed to evaluate multitude of points placed on two dimensional plane in order to calculate and present a solution comprised of the shortest distance found between two points in the plane. Each point would be given a set of coordinates that would correspond with x-axis and y-axis and therefore provide a location on the plane. The algorithm would evaluate the distance between any two points using Pythagorean Theorem formula.

Application

The Closest Pair of Points algorithm could be adapted for similarity search between two nodes in the social network when characteristics of the node would be assigned numerical index values and be used as coordinates. The similarity search could also be used in cluster analysis, computational geometry, and chemical similarity analysis.

Description of two algorithms

A. Brute Force :

- The algorithm finds distances between all pairs of points and chooses minimal distance.
- It iterates through entire array twice using two nested loops to achieve desired computation.
- Computational time is $T(n) = O(n^2)$

```
class BruteForce
{
    public BruteForce(List<Point> points, System.IO.StreamWriter file)
    {
        Segment result = bruteForce(points);
        System.Console.WriteLine("    Point 1 : (" + result.P1.x.ToString() + ", " + result.P1.y.ToString() + ")");
        System.Console.WriteLine("    Point 2 : (" + result.P2.x.ToString() + ", " + result.P2.y.ToString() + ")");
        System.Console.WriteLine("    Shortest distance : " + result.Length().ToString());
        file.WriteLine("    Shortest distance : " + result.Length().ToString());
    }

    public Segment bruteForce(List<Point> points)
    {
        float shortestDistance = (float)Math.Sqrt(Math.Pow(points[0].y - points[1].y, 2.00) + Math.Pow(points[0].x - points[1].x, 2.00));
        Segment result = new Segment(points[0], points[1]);

        for (int i = 0; i < points.Count; i++)
        {
            for (int j = i + 1; j < points.Count; j++)
            {
                if (Math.Sqrt(Math.Pow(points[j].y - points[i].y, 2.00) + Math.Pow(points[j].x - points[i].x, 2.00)) < shortestDistance)
                {
                    shortestDistance = (float)Math.Sqrt(Math.Pow(points[j].y - points[i].y, 2.00) + Math.Pow(points[j].x - points[i].x, 2.00));
                    result = new Segment(points[i], points[j]);
                }
            }
        }
        return result;
    }
}

class Segment
{
    public Segment(Point p1, Point p2)
    {
        P1 = p1;
        P2 = p2;
    }

    public Point P1;
    public Point P2;

    public float Length()
    {
        return (float) Math.Sqrt(LengthSquared());
    }

    public double LengthSquared()
    {
        double result = ((P1.x - P2.x) * (P1.x - P2.x) + (P1.y - P2.y) * (P1.y - P2.y));
        return result;
    }
}

class Point
{
    public double x { get; set; }
    public double y { get; set; }

    public Point(double xCoordinate, double yCoordinate)
    {
        x = xCoordinate;
        y = yCoordinate;
    }
}
```

B. Divide and Conquer :

- The algorithm sorts all points by x coordinate and divides all points in two halves.
- It reclusively finds the smallest distance in both arrays and compares two smallest distances from each half to distances found in the middle area.
- Computational time is T(n) = 2 T(n / 2) + O (n) = O (n lg n)

```
class DivideAndConquer
{
    public DivideAndConquer(List<Point> points, System.IO.StreamWriter file)
    {
        points = points.OrderBy(o => o.x).ToList();
        Segment result = divide(points);
        System.Console.WriteLine("   Point 1 : (" + result.P1.x.ToString() + ", " + result.P1.y.ToString() + ")");
        System.Console.WriteLine("   Point 2 : (" + result.P2.x.ToString() + ", " + result.P2.y.ToString() + ")");
        System.Console.WriteLine("   Shortest distance : " + result.Length().ToString());
        file.WriteLine("   Shortest distance : " + result.Length().ToString());
    }

    public Segment divide(List<Point> points)
    {

        int count = points.Count;
        if (count <= 4) return bruteForce(points);

        List<Point> leftSegment = points.Take(count / 2).ToList();
        Segment leftResult = divide(leftSegment);

        List<Point> rightSegment = points.Skip(count / 2).ToList();
        Segment rightResult = divide(rightSegment);

        Segment result = rightResult.Length() < leftResult.Length() ? rightResult : leftResult;

        double midSegment = leftSegment.Last().x;
        float bandWidth = result.Length();
        var segmentX = points.Where(p => Math.Abs(midSegment - p.x) <= bandWidth);
        var segmentY = segmentX.OrderBy(p => p.y).ToArray();

        int iLast = segmentY.Length - 1;
        for (int i = 0; i < iLast; i++)
        {
            Point pointLow = segmentY[i];

            for (int j = i + 1; j <= iLast; j++)
            {
                Point pointUp = segmentY[j];
                if ((pointUp.y - pointLow.y) >= result.Length())
                    break;
                if (new Segment(pointLow, pointUp).Length() < result.Length()) result = new Segment(pointLow, pointUp);
            }
        }
        return result;
    }

    public Segment bruteForce(List<Point> points)
    {
        float shortestDistance = (float)Math.Sqrt(Math.Pow(points[0].y - points[1].y, 2.00) + Math.Pow(points[0].x - points[1].x, 2.00));
        Segment result = new Segment(points[0], points[1]);

        for (int i = 0; i < points.Count; i++)
        {
            for (int j = i + 1; j < points.Count; j++)
            {
                if (Math.Sqrt(Math.Pow(points[j].y - points[i].y, 2.00) + Math.Pow(points[j].x - points[i].x, 2.00)) < shortestDistance)
                {
                    shortestDistance = (float)Math.Sqrt(Math.Pow(points[j].y - points[i].y, 2.00) + Math.Pow(points[j].x - points[i].x, 2.00));
                    result = new Segment(points[i], points[j]);
                }
            }
        }
        return result;
    }
}

class Segment
{
    public Segment(Point p1, Point p2)
    {
        P1 = p1;
        P2 = p2;
    }

    public Point P1;
    public Point P2;

    public float Length()
    {
        return (float) Math.Sqrt(LengthSquared());
    }

    public double LengthSquared()
    {
        double result = ((P1.x - P2.x) * (P1.x - P2.x) + (P1.y - P2.y) * (P1.y - P2.y));
        return result;
    }
}

class Point
{
    public double x { get; set; }
    public double y { get; set; }

    public Point(double xCoordinate, double yCoordinate)
    {
        x = xCoordinate;
        y = yCoordinate;
    }
}
```

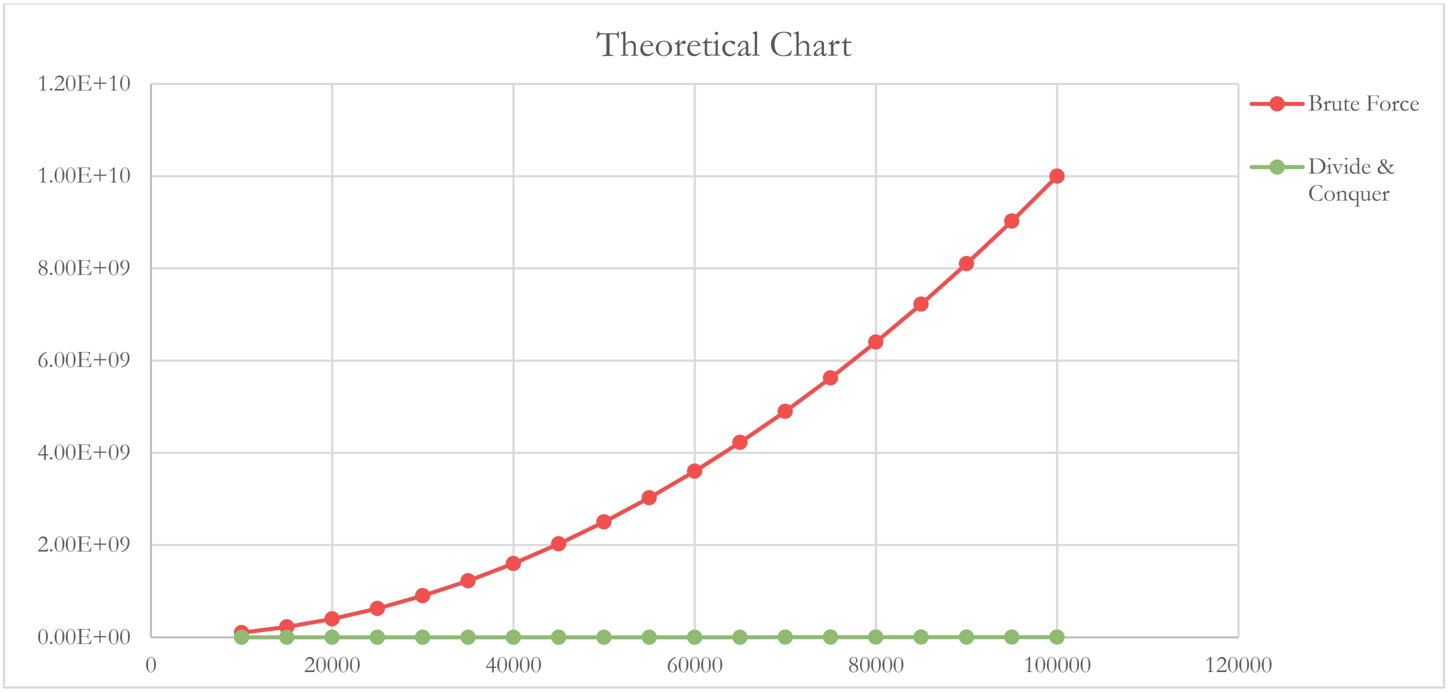
Experiment description

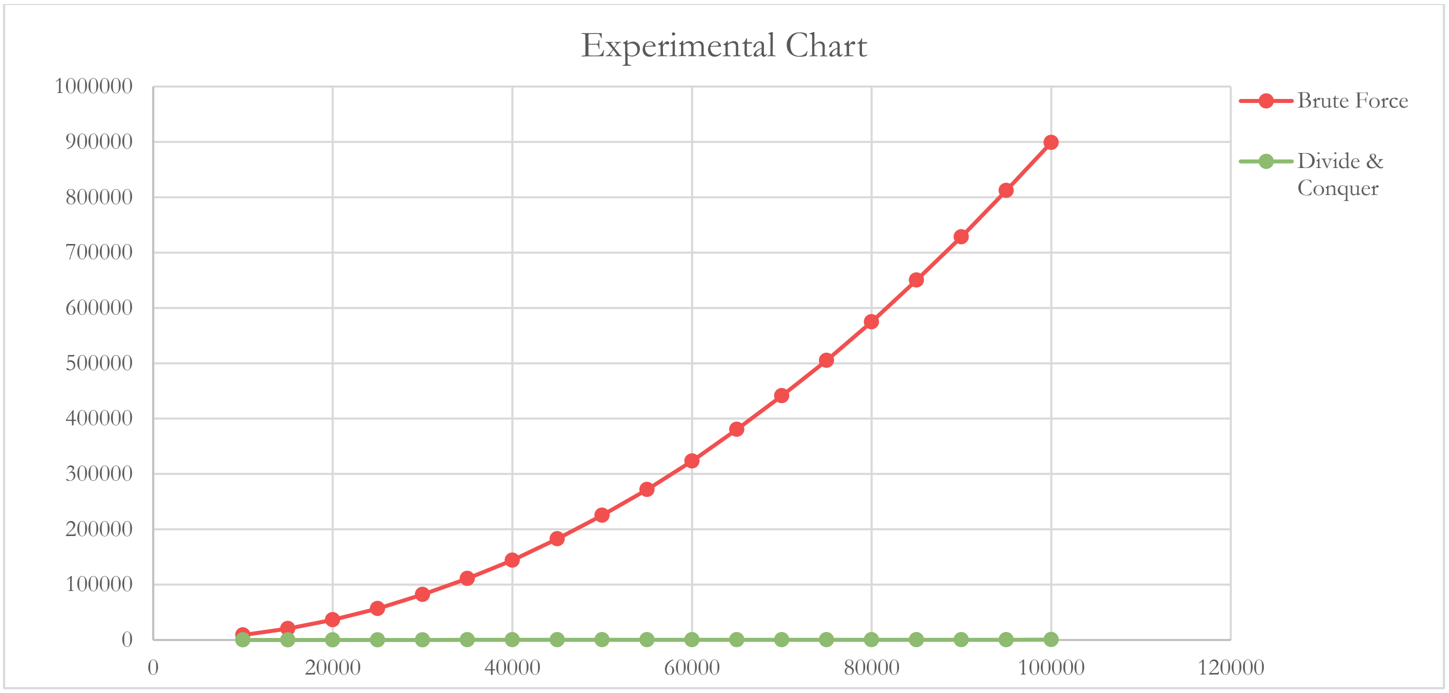
Both algorithms were implemented using C Sharp language and results were saved to text file. The program was built to create 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000, 50000, 55000, 60000, 65000, 70000, 75000, 80000, 85000, 90000, 95000, and 100000 individual points on the plane of size 10000000 x 10000000 where points were placed at random. There were five runs conducted for each n value to determine average running time for each value of n in both Brute Force and Divide & Conquer algorithms. The results were displayed in total millisecond ticks and (hours : minutes : seconds : milliseconds) time format.

	Run 1 (msec)		Run 2 (msec)		Run 3 (msec)		Run 4 (msec)		Run 5 (msec)		Average (msec)	
n	BF	DC	BF	DC	BF	DC	BF	DC	BF	DC	BF	DC
10000	9074	41	9051	36	9076	33	9029	33	9016	33	9049	35
15000	20381	49	20356	49	20342	47	20388	47	20303	49	20354	48
20000	36252	76	36330	73	36293	70	36194	71	36270	70	36268	72
25000	56667	91	56648	86	56516	85	56519	85	56765	85	56623	86
30000	82225	106	81876	98	81830	98	82000	95	81782	105	81943	100
35000	110630	130	111034	122	111172	121	110884	125	110607	120	110865	124
40000	143999	155	143974	150	143918	148	143939	150	143940	147	143954	150
45000	182579	176	182504	168	182647	164	182650	175	182588	175	182594	172
50000	225848	198	225387	178	225205	178	224967	189	224933	190	225268	187
55000	271420	218	271418	201	271780	198	271778	209	271692	195	271618	204
60000	323182	226	323178	216	323060	212	323212	220	323659	217	323258	218
65000	379482	241	379794	230	380788	224	380752	229	380908	223	380345	229
70000	441384	291	440808	267	440974	265	440769	262	441705	266	441128	270
75000	505014	315	505230	293	504976	293	504970	289	505571	294	505152	297
80000	575569	349	575639	320	574551	322	574519	319	574287	318	574913	326
85000	649611	376	650898	337	651194	340	649944	334	650240	341	650377	346
90000	728849	387	728213	354	729570	352	728487	350	726800	354	728384	359
95000	811728	412	812451	378	812572	373	811878	368	812062	375	812138	381
100000	900444	436	905611	394	895791	395	896070	392	896313	392	898846	402

The approximation of constant C was calculated by experimental run / theoretical run.

	Theorethical Run		Experimental Run (msec)		Approximation of C		Standard Deviation of C	
n	BF	DC	BF	DC	BF	DC	BF	DC
10000	100000000	132877	9049	35	0.0000905	0.0002649	0.36%	10.13%
15000	225000000	208090	20354	48	0.0000905	0.0002316	0.33%	3.70%
20000	400000000	285754	36268	72	0.0000907	0.0002520	0.56%	4.75%
25000	625000000	365241	56623	86	0.0000906	0.0002366	0.48%	1.66%
30000	900000000	446180	81943	100	0.0000910	0.0002250	0.98%	6.45%
35000	1225000000	528327	110865	124	0.0000905	0.0002339	0.38%	2.74%
40000	1600000000	611508	143954	150	0.0000900	0.0002453	0.21%	1.98%
45000	2025000000	695594	182594	172	0.0000902	0.0002467	0.01%	2.56%
50000	2500000000	780482	225268	187	0.0000901	0.0002391	0.06%	0.61%
55000	3025000000	866093	271618	204	0.0000898	0.0002358	0.41%	1.98%
60000	3600000000	952360	323258	218	0.0000898	0.0002291	0.41%	4.75%
65000	4225000000	1039230	380345	229	0.0000900	0.0002207	0.16%	8.23%
70000	4900000000	1126655	441128	270	0.0000900	0.0002398	0.15%	0.30%
75000	5625000000	1214595	505152	297	0.0000898	0.0002444	0.40%	1.59%
80000	6400000000	1303017	574913	326	0.0000898	0.0002499	0.37%	3.88%
85000	7225000000	1391890	650377	346	0.0000900	0.0002483	0.16%	3.22%
90000	8100000000	1481187	728384	359	0.0000899	0.0002426	0.27%	0.87%
95000	9025000000	1570886	812138	381	0.0000900	0.0002427	0.19%	0.88%
100000	10000000000	1660964	898846	402	0.0000899	0.0002419	0.31%	0.57%
				Average	0.0000902	0.0002405	0.33%	3.20%

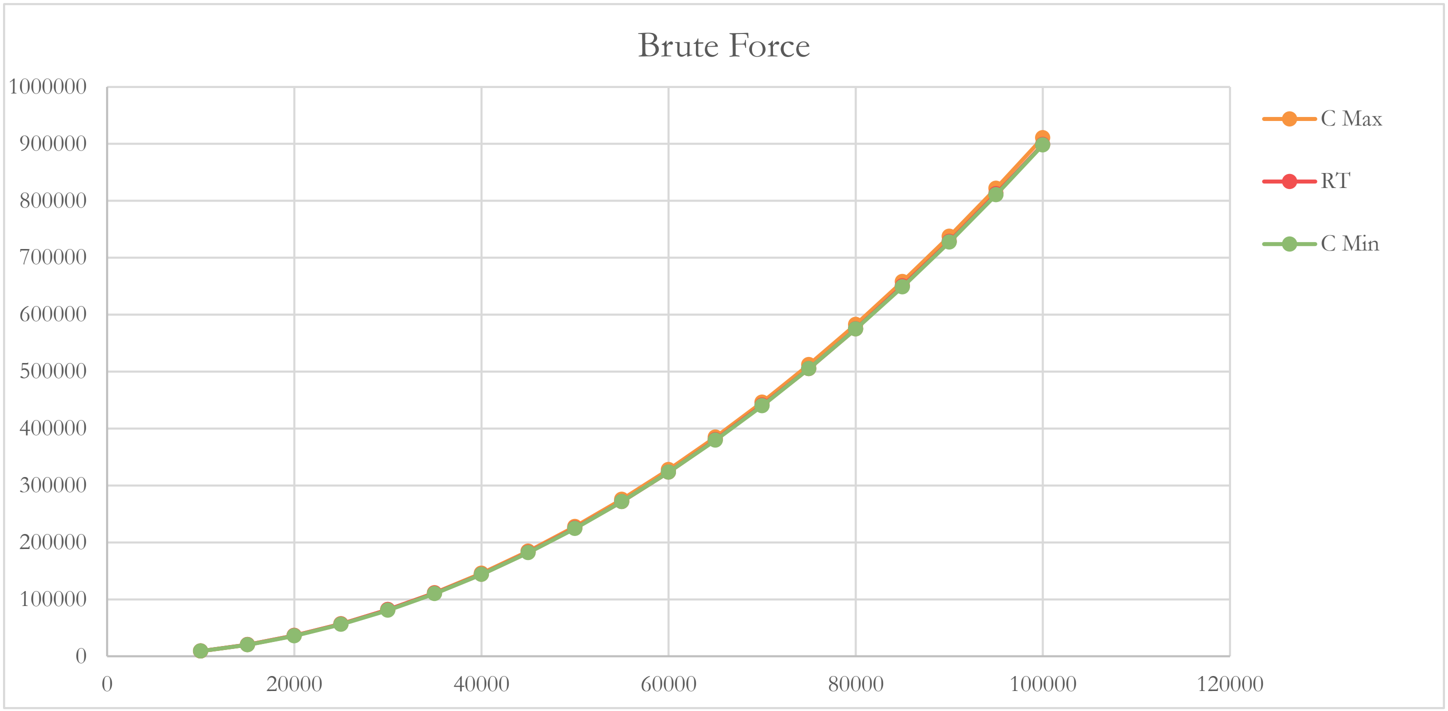




From both times and the graphs we can see the massive improvement in run time of Divide and Conquer algorithm compared to Brute Force. With n equal to 100000 points Brute Force algorithm took 898846 ticks on average which is around 15 minutes while Divide and Conquer algorithm took 402 ticks on average which is less than half a second.

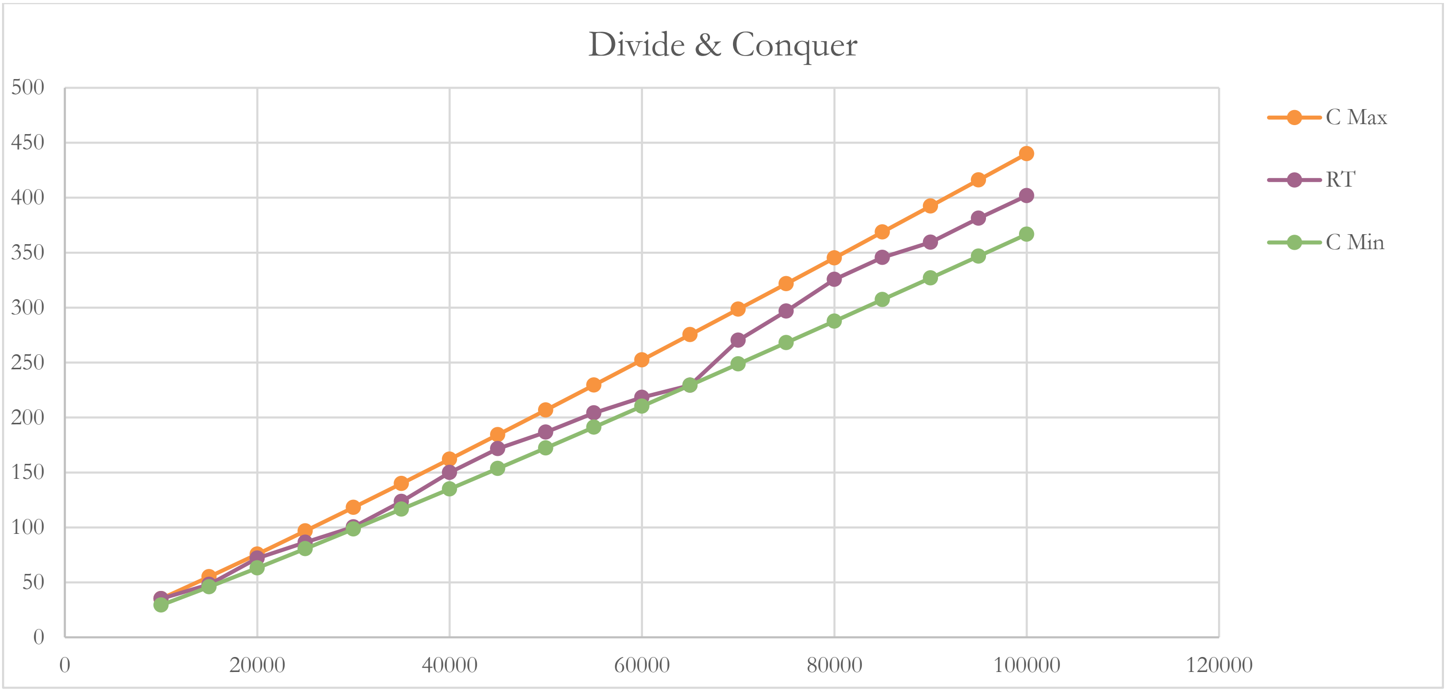
A. Brute Force theoretical results were calculated and the runtime for each calculation was recorded in milliseconds for n values with calculated constant c = 0.0000902 , c_{min} = 0.00008979 , and c_{max} = 0.0009105

	Theoretical	Experimental Run Time Brute Force		
n	RT	C_{min} 0.000089791	BF RT	C_{max} 0.000091047
10000	1.00E+08	8979	9049	9105
15000	2.25E+08	20203	20354	20486
20000	4.00E+08	35916	36268	36419
25000	6.25E+08	56119	56623	56905
30000	9.00E+08	80812	81943	81943
35000	1.23E+09	109994	110865	111533
40000	1.60E+09	143666	143954	145676
45000	2.03E+09	181827	182594	184371
50000	2.50E+09	224477	225268	227618
55000	3.03E+09	271618	271618	275418
60000	3.60E+09	323247	323258	327770
65000	4.23E+09	379367	380345	384675
70000	4.90E+09	439976	441128	446132
75000	5.63E+09	505074	505152	512141
80000	6.40E+09	574662	574913	582703
85000	7.23E+09	648740	650377	657817
90000	8.10E+09	727307	728384	737483
95000	9.03E+09	810363	812138	821702
100000	1.00E+10	897909	898846	910473



B. Divide and Conquer theoretical results were calculated and the runtime for each calculation was recorded in milliseconds for n values with calculated constant $c = 0.0002405$, $c_{min} = 0.00022074$, and $c_{max} = 0.00026491$

	Theoretical	Experimental Run Time Divide & Conquer		
n	RT	C_{min} 0.000220740	DC RT	C_{max} 0.000264906
10000	1.33E+05	29	35	35
15000	2.08E+05	46	48	55
20000	2.86E+05	63	72	76
25000	3.65E+05	81	86	97
30000	4.46E+05	98	100	118
35000	5.28E+05	117	124	140
40000	6.12E+05	135	150	162
45000	6.96E+05	154	172	184
50000	7.80E+05	172	187	207
55000	8.66E+05	191	204	229
60000	9.52E+05	210	218	252
65000	1.04E+06	229	229	275
70000	1.13E+06	249	270	298
75000	1.21E+06	268	297	322
80000	1.30E+06	288	326	345
85000	1.39E+06	307	346	369
90000	1.48E+06	327	359	392
95000	1.57E+06	347	381	416
100000	1.66E+06	367	402	440



Results Captured

```
Calcualtions for 10000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 322.1699
   Function run time : 00:00:09.0740000
   Milisecs run time : 9074
   Shortest distance with Divide & Conquer
   Shortest distance : 322.1699
   Function run time : 00:00:00.0410000
   Milisecs run time : 41
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 322.1699
   Function run time : 00:00:09.0510000
   Milisecs run time : 9051
   Shortest distance with Divide & Conquer
   Shortest distance : 322.1699
   Function run time : 00:00:00.0360000
   Milisecs run time : 36
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 322.1699
   Function run time : 00:00:09.0760000
   Milisecs run time : 9076
   Shortest distance with Divide & Conquer
   Shortest distance : 322.1699
   Function run time : 00:00:00.0330000
   Milisecs run time : 33
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 322.1699
   Function run time : 00:00:09.0290000
   Milisecs run time : 9029
   Shortest distance with Divide & Conquer
   Shortest distance : 322.1699
   Function run time : 00:00:00.0330000
   Milisecs run time : 33
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 322.1699
   Function run time : 00:00:09.0160000
   Milisecs run time : 9016
   Shortest distance with Divide & Conquer
   Shortest distance : 322.1699
   Function run time : 00:00:00.0330000
   Milisecs run time : 33

Calcualtions for 15000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 359.2356
   Function run time : 00:00:20.3810000
   Milisecs run time : 20381
   Shortest distance with Divide & Conquer
   Shortest distance : 359.2356
   Function run time : 00:00:00.0490000
   Milisecs run time : 49
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 359.2356
   Function run time : 00:00:20.3560000
   Milisecs run time : 20356
   Shortest distance with Divide & Conquer
   Shortest distance : 359.2356
   Function run time : 00:00:00.0490000
   Milisecs run time : 49
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 359.2356
   Function run time : 00:00:20.3420000
   Milisecs run time : 20342
   Shortest distance with Divide & Conquer
   Shortest distance : 359.2356
   Function run time : 00:00:00.0470000
   Milisecs run time : 47
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 359.2356
   Function run time : 00:00:20.3880000
   Milisecs run time : 20388
   Shortest distance with Divide & Conquer
   Shortest distance : 359.2356
   Function run time : 00:00:00.0470000
   Milisecs run time : 47
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 359.2356
   Function run time : 00:00:20.3030000
   Milisecs run time : 20303
   Shortest distance with Divide & Conquer
   Shortest distance : 359.2356
   Function run time : 00:00:00.0490000
   Milisecs run time : 49

Calcualtions for 20000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 335.429
   Function run time : 00:00:36.2520000
   Milisecs run time : 36252
   Shortest distance with Divide & Conquer
   Shortest distance : 335.429
   Function run time : 00:00:00.0760000
   Milisecs run time : 76
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 335.429
   Function run time : 00:00:36.3300000
   Milisecs run time : 36330
   Shortest distance with Divide & Conquer
   Shortest distance : 335.429
   Function run time : 00:00:00.0730000
   Milisecs run time : 73
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 335.429
   Function run time : 00:00:36.2930000
   Milisecs run time : 36293
   Shortest distance with Divide & Conquer
   Shortest distance : 335.429
   Function run time : 00:00:00.0700000
   Milisecs run time : 70
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 335.429
   Function run time : 00:00:36.1940000
   Milisecs run time : 36194
   Shortest distance with Divide & Conquer
   Shortest distance : 335.429
   Function run time : 00:00:00.0710000
   Milisecs run time : 71
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 335.429
   Function run time : 00:00:36.2700000
   Milisecs run time : 36270
   Shortest distance with Divide & Conquer
   Shortest distance : 335.429
   Function run time : 00:00:00.0700000
   Milisecs run time : 70

Calcualtions for 25000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 280.6194
   Function run time : 00:00:56.6670000
   Milisecs run time : 56667
   Shortest distance with Divide & Conquer
   Shortest distance : 280.6194
   Function run time : 00:00:00.0910000
   Milisecs run time : 91
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 280.6194
   Function run time : 00:00:56.6480000
   Milisecs run time : 56648
   Shortest distance with Divide & Conquer
   Shortest distance : 280.6194
   Function run time : 00:00:00.0860000
   Milisecs run time : 86
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 280.6194
   Function run time : 00:00:56.5160000
   Milisecs run time : 56516
   Shortest distance with Divide & Conquer
   Shortest distance : 280.6194
   Function run time : 00:00:00.0850000
   Milisecs run time : 85
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 280.6194
   Function run time : 00:00:56.5190000
   Milisecs run time : 56519
   Shortest distance with Divide & Conquer
   Shortest distance : 280.6194
   Function run time : 00:00:00.0850000
   Milisecs run time : 85
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 280.6194
   Function run time : 00:00:56.7650000
   Milisecs run time : 56765
   Shortest distance with Divide & Conquer
   Shortest distance : 280.6194
   Function run time : 00:00:00.0850000
   Milisecs run time : 85

Calcualtions for 30000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 290.2854
   Function run time : 00:01:22.2250000
   Milisecs run time : 82225
   Shortest distance with Divide & Conquer
   Shortest distance : 290.2854
   Function run time : 00:00:00.1060000
   Milisecs run time : 106
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 290.2854
   Function run time : 00:01:21.8760000
   Milisecs run time : 81876
   Shortest distance with Divide & Conquer
   Shortest distance : 290.2854
   Function run time : 00:00:00.0980000
   Milisecs run time : 98
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 290.2854
   Function run time : 00:01:21.8300000
   Milisecs run time : 81830
   Shortest distance with Divide & Conquer
   Shortest distance : 290.2854
   Function run time : 00:00:00.0980000
   Milisecs run time : 98
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 290.2854
   Function run time : 00:01:22
   Milisecs run time : 82000
   Shortest distance with Divide & Conquer
   Shortest distance : 290.2854
   Function run time : 00:00:00.0950000
   Milisecs run time : 95
```

```
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 290.2854
   Function run time : 00:01:21.7820000
   Milisecs run time : 81782
   Shortest distance with Divide & Conquer
   Shortest distance : 290.2854
   Function run time : 00:00:00.1050000
   Milisecs run time : 105

Calcualtions for 35000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 110.4133
   Function run time : 00:01:50.6300000
   Milisecs run time : 110630
   Shortest distance with Divide & Conquer
   Shortest distance : 110.4133
   Function run time : 00:00:00.1300000
   Milisecs run time : 130
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 110.4133
   Function run time : 00:01:51.0340000
   Milisecs run time : 111034
   Shortest distance with Divide & Conquer
   Shortest distance : 110.4133
   Function run time : 00:00:00.1220000
   Milisecs run time : 122
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 110.4133
   Function run time : 00:01:51.1720000
   Milisecs run time : 111172
   Shortest distance with Divide & Conquer
   Shortest distance : 110.4133
   Function run time : 00:00:00.1210000
   Milisecs run time : 121
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 110.4133
   Function run time : 00:01:50.8840000
   Milisecs run time : 110884
   Shortest distance with Divide & Conquer
   Shortest distance : 110.4133
   Function run time : 00:00:00.1250000
   Milisecs run time : 125
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 110.4133
   Function run time : 00:01:50.6070000
   Milisecs run time : 110607
   Shortest distance with Divide & Conquer
   Shortest distance : 110.4133
   Function run time : 00:00:00.1200000
   Milisecs run time : 120

Calcualtions for 40000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 341.3802
   Function run time : 00:02:23.9990000
   Milisecs run time : 143999
   Shortest distance with Divide & Conquer
   Shortest distance : 341.3802
   Function run time : 00:00:00.1550000
   Milisecs run time : 155
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 341.3802
   Function run time : 00:02:23.9740000
   Milisecs run time : 143974
   Shortest distance with Divide & Conquer
   Shortest distance : 341.3802
   Function run time : 00:00:00.1500000
   Milisecs run time : 150
```

```
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 341.3802
   Function run time : 00:02:23.9180000
   Milisecs run time : 143918
   Shortest distance with Divide & Conquer
   Shortest distance : 341.3802
   Function run time : 00:00:00.1480000
   Milisecs run time : 148
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 341.3802
   Function run time : 00:02:23.9390000
   Milisecs run time : 143939
   Shortest distance with Divide & Conquer
   Shortest distance : 341.3802
   Function run time : 00:00:00.1500000
   Milisecs run time : 150
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 341.3802
   Function run time : 00:02:23.9400000
   Milisecs run time : 143940
   Shortest distance with Divide & Conquer
   Shortest distance : 341.3802
   Function run time : 00:00:00.1470000
   Milisecs run time : 147

Calcualtions for 45000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 46.89661
   Function run time : 00:03:02.5790000
   Milisecs run time : 182579
   Shortest distance with Divide & Conquer
   Shortest distance : 46.89661
   Function run time : 00:00:00.1760000
   Milisecs run time : 176
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 46.89661
   Function run time : 00:03:02.5040000
   Milisecs run time : 182504
   Shortest distance with Divide & Conquer
   Shortest distance : 46.89661
   Function run time : 00:00:00.1680000
   Milisecs run time : 168
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 46.89661
   Function run time : 00:03:02.6470000
   Milisecs run time : 182647
   Shortest distance with Divide & Conquer
   Shortest distance : 46.89661
   Function run time : 00:00:00.1640000
   Milisecs run time : 164
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 46.89661
   Function run time : 00:03:02.6500000
   Milisecs run time : 182650
   Shortest distance with Divide & Conquer
   Shortest distance : 46.89661
   Function run time : 00:00:00.1750000
   Milisecs run time : 175
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 46.89661
   Function run time : 00:03:02.5880000
   Milisecs run time : 182588
   Shortest distance with Divide & Conquer
   Shortest distance : 46.89661
   Function run time : 00:00:00.1750000
   Milisecs run time : 175

Calcualtions for 50000 points
```

```
Run # 1
   Shortest distance with Brute Force
   Shortest distance : 78.34741
   Function run time : 00:03:45.8480000
   Milisecs run time : 225848
   Shortest distance with Divide & Conquer
   Shortest distance : 78.34741
   Function run time : 00:00:00.1980000
   Milisecs run time : 198
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 78.34741
   Function run time : 00:03:45.3870000
   Milisecs run time : 225387
   Shortest distance with Divide & Conquer
   Shortest distance : 78.34741
   Function run time : 00:00:00.1780000
   Milisecs run time : 178
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 78.34741
   Function run time : 00:03:45.2050000
   Milisecs run time : 225205
   Shortest distance with Divide & Conquer
   Shortest distance : 78.34741
   Function run time : 00:00:00.1780000
   Milisecs run time : 178
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 78.34741
   Function run time : 00:03:44.9670000
   Milisecs run time : 224967
   Shortest distance with Divide & Conquer
   Shortest distance : 78.34741
   Function run time : 00:00:00.1890000
   Milisecs run time : 189
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 78.34741
   Function run time : 00:03:44.9330000
   Milisecs run time : 224933
   Shortest distance with Divide & Conquer
   Shortest distance : 78.34741
   Function run time : 00:00:00.1900000
   Milisecs run time : 190

Calcualtions for 55000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 109.9756
   Function run time : 00:04:31.4200000
   Milisecs run time : 271420
   Shortest distance with Divide & Conquer
   Shortest distance : 109.9756
   Function run time : 00:00:00.2180000
   Milisecs run time : 218
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 109.9756
   Function run time : 00:04:31.4180000
   Milisecs run time : 271418
   Shortest distance with Divide & Conquer
   Shortest distance : 109.9756
   Function run time : 00:00:00.2010000
   Milisecs run time : 201
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 109.9756
   Function run time : 00:04:31.7800000
   Milisecs run time : 271780
   Shortest distance with Divide & Conquer
   Shortest distance : 109.9756
   Function run time : 00:00:00.1980000
   Milisecs run time : 198
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 109.9756
   Function run time : 00:04:31.7780000
   Milisecs run time : 271778
   Shortest distance with Divide & Conquer
   Shortest distance : 109.9756
   Function run time : 00:00:00.2090000
   Milisecs run time : 209
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 109.9756
   Function run time : 00:04:31.6920000
   Milisecs run time : 271692
   Shortest distance with Divide & Conquer
   Shortest distance : 109.9756
   Function run time : 00:00:00.1950000
   Milisecs run time : 195

Calcualtions for 60000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 74.03804
   Function run time : 00:05:23.1820000
   Milisecs run time : 323182
   Shortest distance with Divide & Conquer
   Shortest distance : 74.03804
   Function run time : 00:00:00.2260000
   Milisecs run time : 226
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 74.03804
   Function run time : 00:05:23.1780000
   Milisecs run time : 323178
   Shortest distance with Divide & Conquer
   Shortest distance : 74.03804
   Function run time : 00:00:00.2160000
   Milisecs run time : 216
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 74.03804
   Function run time : 00:05:23.0600000
   Milisecs run time : 323060
   Shortest distance with Divide & Conquer
   Shortest distance : 74.03804
   Function run time : 00:00:00.2120000
   Milisecs run time : 212
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 74.03804
   Function run time : 00:05:23.2120000
   Milisecs run time : 323212
   Shortest distance with Divide & Conquer
   Shortest distance : 74.03804
   Function run time : 00:00:00.2200000
   Milisecs run time : 220
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 74.03804
   Function run time : 00:05:23.6590000
   Milisecs run time : 323659
   Shortest distance with Divide & Conquer
   Shortest distance : 74.03804
   Function run time : 00:00:00.2170000
   Milisecs run time : 217

Calcualtions for 65000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 111.1195
   Function run time : 00:06:19.4820000
   Milisecs run time : 379482
   Shortest distance with Divide & Conquer
   Shortest distance : 111.1195
   Function run time : 00:00:00.2410000
   Milisecs run time : 241
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 111.1195
   Function run time : 00:06:19.7940000
   Milisecs run time : 379794
   Shortest distance with Divide & Conquer
   Shortest distance : 111.1195
   Function run time : 00:00:00.2300000
   Milisecs run time : 230
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 111.1195
   Function run time : 00:06:20.7880000
   Milisecs run time : 380788
   Shortest distance with Divide & Conquer
   Shortest distance : 111.1195
   Function run time : 00:00:00.2240000
   Milisecs run time : 224
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 111.1195
   Function run time : 00:06:20.7520000
   Milisecs run time : 380752
   Shortest distance with Divide & Conquer
   Shortest distance : 111.1195
   Function run time : 00:00:00.2290000
   Milisecs run time : 229
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 111.1195
   Function run time : 00:06:20.9080000
   Milisecs run time : 380908
   Shortest distance with Divide & Conquer
   Shortest distance : 111.1195
   Function run time : 00:00:00.2230000
   Milisecs run time : 223

Calcualtions for 70000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 192.2958
   Function run time : 00:07:21.3840000
   Milisecs run time : 441384
   Shortest distance with Divide & Conquer
   Shortest distance : 192.2958
   Function run time : 00:00:00.2910000
   Milisecs run time : 291
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 192.2958
   Function run time : 00:07:20.8080000
   Milisecs run time : 440808
   Shortest distance with Divide & Conquer
   Shortest distance : 192.2958
   Function run time : 00:00:00.2670000
   Milisecs run time : 267
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 192.2958
   Function run time : 00:07:20.9740000
   Milisecs run time : 440974
   Shortest distance with Divide & Conquer
   Shortest distance : 192.2958
   Function run time : 00:00:00.2650000
   Milisecs run time : 265
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 192.2958
   Function run time : 00:07:20.7690000
   Milisecs run time : 440769
   Shortest distance with Divide & Conquer
   Shortest distance : 192.2958
   Function run time : 00:00:00.2620000
   Milisecs run time : 262
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 192.2958
   Function run time : 00:07:21.7050000
   Milisecs run time : 441705
   Shortest distance with Divide & Conquer
   Shortest distance : 192.2958
   Function run time : 00:00:00.2660000
   Milisecs run time : 266

Calcualtions for 75000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 132.993
   Function run time : 00:08:25.0140000
   Milisecs run time : 505014
   Shortest distance with Divide & Conquer
   Shortest distance : 132.993
   Function run time : 00:00:00.3150000
   Milisecs run time : 315
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 132.993
   Function run time : 00:08:25.2300000
   Milisecs run time : 505230
   Shortest distance with Divide & Conquer
   Shortest distance : 132.993
   Function run time : 00:00:00.2930000
   Milisecs run time : 293
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 132.993
   Function run time : 00:08:24.9760000
   Milisecs run time : 504976
   Shortest distance with Divide & Conquer
   Shortest distance : 132.993
   Function run time : 00:00:00.2930000
   Milisecs run time : 293
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 132.993
   Function run time : 00:08:24.9700000
   Milisecs run time : 504970
   Shortest distance with Divide & Conquer
   Shortest distance : 132.993
   Function run time : 00:00:00.2890000
   Milisecs run time : 289
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 132.993
   Function run time : 00:08:25.5710000
   Milisecs run time : 505571
   Shortest distance with Divide & Conquer
   Shortest distance : 132.993
   Function run time : 00:00:00.2940000
   Milisecs run time : 294

Calcualtions for 80000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 24.42979
   Function run time : 00:09:35.5690000
   Milisecs run time : 575569
   Shortest distance with Divide & Conquer
   Shortest distance : 24.42979
   Function run time : 00:00:00.3490000
   Milisecs run time : 349
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 24.42979
   Function run time : 00:09:35.6350000
   Milisecs run time : 575635
   Shortest distance with Divide & Conquer
   Shortest distance : 24.42979
   Function run time : 00:00:00.3200000
   Milisecs run time : 320
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 24.42979
   Function run time : 00:09:34.5510000
   Milisecs run time : 574551
   Shortest distance with Divide & Conquer
   Shortest distance : 24.42979
   Function run time : 00:00:00.3220000
   Milisecs run time : 322
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 24.42979
   Function run time : 00:09:34.5190000
   Milisecs run time : 574519
   Shortest distance with Divide & Conquer
   Shortest distance : 24.42979
   Function run time : 00:00:00.3190000
   Milisecs run time : 319
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 24.42979
   Function run time : 00:09:34.2870000
   Milisecs run time : 574287
   Shortest distance with Divide & Conquer
   Shortest distance : 24.42979
   Function run time : 00:00:00.3180000
   Milisecs run time : 318

Calcualtions for 85000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 39.93435
   Function run time : 00:10:49.6110000
   Milisecs run time : 649611
   Shortest distance with Divide & Conquer
   Shortest distance : 39.93435
   Function run time : 00:00:00.3760000
   Milisecs run time : 376
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 39.93435
   Function run time : 00:10:50.8980000
   Milisecs run time : 650898
   Shortest distance with Divide & Conquer
   Shortest distance : 39.93435
   Function run time : 00:00:00.3370000
   Milisecs run time : 337
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 39.93435
   Function run time : 00:10:51.1940000
   Milisecs run time : 651194
   Shortest distance with Divide & Conquer
   Shortest distance : 39.93435
   Function run time : 00:00:00.3400000
   Milisecs run time : 340
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 39.93435
   Function run time : 00:10:49.9440000
   Milisecs run time : 649944
   Shortest distance with Divide & Conquer
   Shortest distance : 39.93435
   Function run time : 00:00:00.3340000
   Milisecs run time : 334
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 39.93435
   Function run time : 00:10:50.2400000
   Milisecs run time : 650240
   Shortest distance with Divide & Conquer
   Shortest distance : 39.93435
   Function run time : 00:00:00.3410000
   Milisecs run time : 341

Calcualtions for 90000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 83.07877
   Function run time : 00:12:08.8490000
   Milisecs run time : 728849
   Shortest distance with Divide & Conquer
   Shortest distance : 83.07877
   Function run time : 00:00:00.3870000
   Milisecs run time : 387
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 83.07877
   Function run time : 00:12:08.2130000
   Milisecs run time : 728213
   Shortest distance with Divide & Conquer
   Shortest distance : 83.07877
   Function run time : 00:00:00.3540000
   Milisecs run time : 354
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 83.07877
   Function run time : 00:12:09.5700000
   Milisecs run time : 729570
   Shortest distance with Divide & Conquer
   Shortest distance : 83.07877
   Function run time : 00:00:00.3520000
   Milisecs run time : 352
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 83.07877
   Function run time : 00:12:08.4870000
   Milisecs run time : 728487
   Shortest distance with Divide & Conquer
   Shortest distance : 83.07877
   Function run time : 00:00:00.3500000
   Milisecs run time : 350
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 83.07877
   Function run time : 00:12:06.8000000
   Milisecs run time : 726800
   Shortest distance with Divide & Conquer
   Shortest distance : 83.07877
   Function run time : 00:00:00.3540000
   Milisecs run time : 354

Calcualtions for 95000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 56.21426
   Function run time : 00:13:31.7280000
   Milisecs run time : 811728
   Shortest distance with Divide & Conquer
   Shortest distance : 56.21426
   Function run time : 00:00:00.4120000
   Milisecs run time : 412
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 56.21426
   Function run time : 00:13:32.4510000
   Milisecs run time : 812451
   Shortest distance with Divide & Conquer
   Shortest distance : 56.21426
   Function run time : 00:00:00.3780000
   Milisecs run time : 378
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 56.21426
   Function run time : 00:13:32.5720000
   Milisecs run time : 812572
   Shortest distance with Divide & Conquer
   Shortest distance : 56.21426
   Function run time : 00:00:00.3730000
   Milisecs run time : 373
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 56.21426
   Function run time : 00:13:31.8780000
   Milisecs run time : 811878
   Shortest distance with Divide & Conquer
   Shortest distance : 56.21426
   Function run time : 00:00:00.3680000
   Milisecs run time : 368
```

```
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 56.21426
   Function run time : 00:13:32.0620000
   Milisecs run time : 812062
   Shortest distance with Divide & Conquer
   Shortest distance : 56.21426
   Function run time : 00:00:00.3750000
   Milisecs run time : 375

Calcualtions for 100000 points

Run # 1
   Shortest distance with Brute Force
   Shortest distance : 44.52962
   Function run time : 00:15:00.4440000
   Milisecs run time : 900444
   Shortest distance with Divide & Conquer
   Shortest distance : 44.52962
   Function run time : 00:00:00.4360000
   Milisecs run time : 436
Run # 2
   Shortest distance with Brute Force
   Shortest distance : 44.52962
   Function run time : 00:15:05.6110000
   Milisecs run time : 905611
   Shortest distance with Divide & Conquer
   Shortest distance : 44.52962
   Function run time : 00:00:00.3940000
   Milisecs run time : 394
Run # 3
   Shortest distance with Brute Force
   Shortest distance : 44.52962
   Function run time : 00:14:55.7910000
   Milisecs run time : 895791
   Shortest distance with Divide & Conquer
   Shortest distance : 44.52962
   Function run time : 00:00:00.3950000
   Milisecs run time : 395
Run # 4
   Shortest distance with Brute Force
   Shortest distance : 44.52962
   Function run time : 00:14:56.0700000
   Milisecs run time : 896070
   Shortest distance with Divide & Conquer
   Shortest distance : 44.52962
   Function run time : 00:00:00.3920000
   Milisecs run time : 392
Run # 5
   Shortest distance with Brute Force
   Shortest distance : 44.52962
   Function run time : 00:14:56.3130000
   Milisecs run time : 896313
   Shortest distance with Divide & Conquer
   Shortest distance : 44.52962
   Function run time : 00:00:00.3920000
   Milisecs run time : 392
```

References

Ge, Q., Wang, H., & Zhu, H. (2006). An improved algorithm for finding the closest pair of points. Journal of Computer Science and Technology, 21(1), 27-31. doi:http://dx.doi.org/10.1007/s11390-006-0027-7

Pereira, J. C., & Lobo, F. G. (2012). An optimized divide-and-conquer algorithm for the closest-pair problem in the planar case. Journal of Computer Science and Technology, 27(4), 891-896. doi:http://dx.doi.org/10.1007/s11390-012-1272-6

Richards, M. (2002). A note concerning the closest point pair algorithm. Information Processing Letters, 82(4), 193-195. doi:10.1016/S0020-0190(01)00268-X

Karim, M. Z., & Akter, N. (2011). Optimum partition parameter of divide-and-conquer algorithm for solving closest-pair problem. International Journal of Computer Science & Information Technology, 3(5), 211-219.

Jiang, M., & Gillespie, J. (2007). Engineering the divide-and-conquer closest pair algorithm. Journal of Computer Science and Technology, 22(4), 532-540. doi:10.1007/s11390-007-9066-y

Closest Pair of Points problem. - Wikipedia. *doi:https://en.wikipedia.org/wiki/Closest_pair_of_points_problem*

Closest Pair of Points. - GeeksforGeeks. *doi:http://www.geeksforgeeks.org/closest-pair-of-points-onlogn-implementation*