Group member: Hao Wang

Pseudo-code:

```
Brute Force:
    def main():
        ReadDataFromFile;
        for i in range(len(Data)):
            for j in range(i+1, len(Data)):
                calculate distance between Data[i], Data[j];
                if distance <= min:
                    min = distance;
                    Result.append(Data[i], Data[j], distance);
        remove element which distance != min in Result;
        write result to file;
        return 0;

DnC:
    def closest_pair(data):
        if(len(data) <= 3):
            calculate minimal distance use brute force;
        else:
            midPoint = len(data) / 2;
            left = data[0,midPoint];
            right = data[midPoint, len(data)];
            min_left = closest_pair(left);
            min_right = closest_pair(right);
            get smaller value of min_left and min_right;

            get left and right limit for cross pair;
            cross = data[left_limit, right_limit];
            cross.sort(base on y value);
            min = closest_cross_pair(cross, min);
            return

    def closest_cross_pair(cross, min):
        for i in range(len(data)):
            for j in range(i+1, len(data)):
                if(data[j][y] - data[i][y]):
                    calculate distance between data[i], data[j];
                else:
                    break;

enhancedDnC:

    def closest_pair(data):
        if(len(data) <= 3):
            calculate minimal distance use brute force;
        else:
            midPoint = len(data) / 2;
            left = data[0,midPoint];
            right = data[midPoint, len(data)];
            min_left = closest_pair(left);
            min_right = closest_pair(right);
            get smaller value of min_left and min_right;
```

```
                get left and right limit for cross pair;
                cross = data[left_limit, right_limit]; //data already sorted
                min = closest_cross_pair(cross, min);
                return

        def closest_cross_pair(cross, min):
            for i in range(len(data)):
                for j in range(i+1, len(data)):
                    if(data[j][y] - data[i][y]):
                        calculate distance between data[i], data[j];
                    else:
                        break;
```

Asymptotic Analysis of run time:

Dnc:

$T(n) = 2T(n/2) + cn\log n$

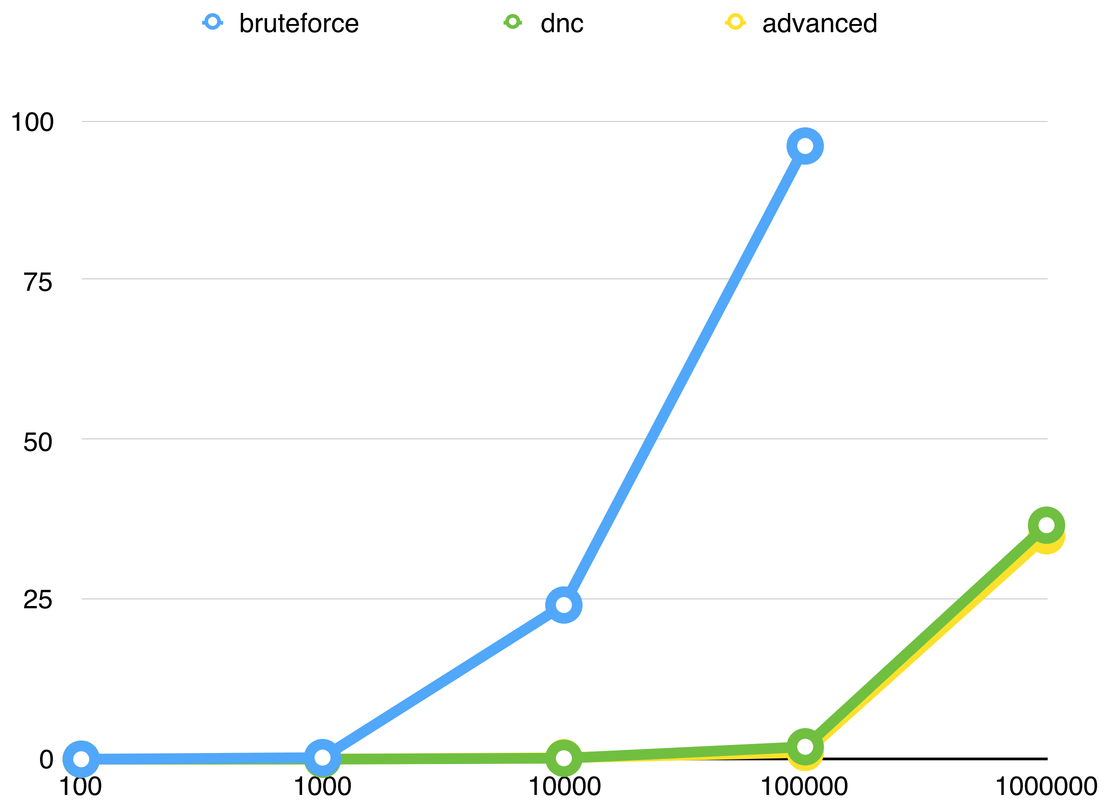
$T(n) = O(n \log^2 n)$

enhanceDnc:

$T(n) = 2T(n/2) + cn$

$T(n) = O(n \log n)$

Plotting:



Interpretation and discussion:

Graph looks agree the run time analysis in last part.