CS 325 - Implementation Assignment 1 - Report

Troy Diaz, Balakrishna Thirumavalavan, Andrew Brandenfels

January 20, 2025

Pseudocode

```
enhanced_dnc (points)
    xSortedPoints = sortByX(points)
    ySortedPoints = sortByY(points)
    return enhanced_dnc_recursive(xSortedPoints, ySortedPoints, 0, len(points) - 1)

enhanced_dnc_recursive (xSortedPoints, ySortedPoints, lowIndex, highIndex)
    % x_m = findMedian(xSortedPoints, x)
    splitIndex = floor((highIndex - lowIndex) / 2)
    x_m = xSortedPoints[splitIndex][0]
    leftHalf = enhanced_dnc_recursive(xSortedPoints, ySortedPoints, lowIndex, splitIndex)
    rightHalf = enhanced_dnc_recursive(xSortedPoints, ySortedPoints, splitIndex + 1, highInd
    d = min(leftHalf.closestPair, rightHalf.closestPair)
    for point in ySortedPoints:
        if x_m - d <= point[0] && point[0] <= x_m + d:
            M.append(point)
    d_m = d
    closestPoints = []
    closestPointsInM = []
    for pointA in M:
        for pointB in M:
            if (pointA[1] - pointB[1] > d):
                goto End_M_Check
            else:
                current_d = computeDistance(pointA, pointB)
                if current_d < d_m:
                    closestPointsInM = [[[pointA],[pointB]]]
                    d_m = current_d
                else if current_d = d_m:
                    closestPointsInM.append([[pointA],[pointB]])
        if d_m < d:
            closestPoints = closestPointsInM
            d = d_m
        else if d_m = d:
            for pointSet in closestPointsInM:
```

```
                closestPoints.append(pointSet)
    End_M_Check
    min_distance = min(d, d_m)
    closestPoints = sort_pairs(closestPoints)
    return min_distance, closestPoints

computeDistance(pointA, pointB)
    return sqrt((pointA[0] - pointB[0])^2 + (pointA[0] - pointB[0])^2)
```

Asymptotic Runtime Analysis