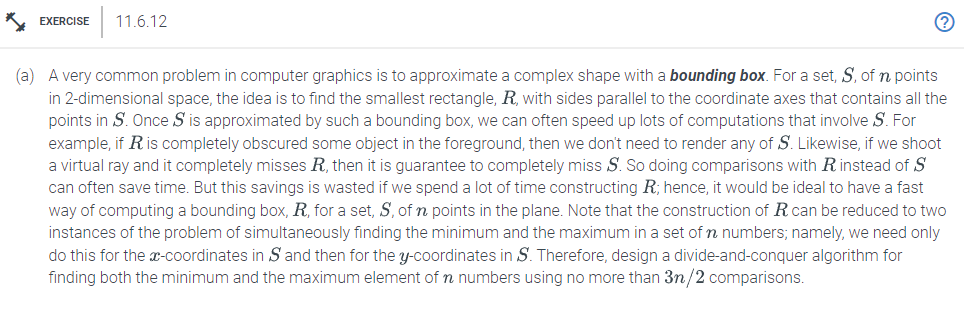
# Michael Chillemi

# 07/20/2023

# CS 590 - Algorithms

# M9.B3: Module 9 Divide-and-Conquer Application Exercises

Problem 11.6.12



Answer:

Here is a divide-and-conquer procedure that uses no more than comparisons to determine both the least and greatest element of n numbers.Return the same element as the minimum and maximum if n=1. If n = 2, compare the two elements and indicate the minimum and maximum values for the smaller element. If n is greater than 2, split the set S into two subsets of nearly similar size, S1 and S2. To do this, we can divide the sorted list in half after sorting the points in S according to their x-coordinates. Using the divide-and-conquer approach, recursively determine the minimum and maximum x-coordinates of S1 and S2.Next the smallest value of S is the smaller of the minimum x-coordinates of S1 and S2. Similar to this, the highest value of S is the highest value of the highest x-coordinates of S1 and S2. Last you would provide S's minimum and maximum. The minimum and maximum y-coordinates of the points in S can be determined using the same procedure, and the bounding box can be calculated by adding the minimum and maximum x- and y-coordinates. Remember that each recursive call divides the set in two and resolves two subproblems of approximately n/2 size while examining the algorithm's running time. As a result, the method performs T(n) = 2T(n/2) + 2 comparisons, where "2" denotes the comparison of the minimum and maximum of the two subsets. The running time of the algorithm is O(n log n), which is quicker than the naive technique that compares every pair of items, which would need O() comparisons. This is demonstrated using the master theorem. As a result, the divide-and-conquer technique uses no more than 3n/2 comparisons to effectively compute the bounding box of a set of n points in 2D space.

To conclude The smallest rectangle that encompasses every point in a set S of n points in 2D space can be used to tackle the problem of approximating a complex shape with a bounding box. Finding the least and maximum x- and y-coordinates of the points in S will solve this issue. This problem can be resolved using a divide-and-conquer algorithm with just 3n/2 comparisons. The minimum and maximum of the set are determined by combining the solutions to two subproblems of approximately n/2 size that were solved as part of the recursive division of the set into two parts. The algorithm's execution time is O(n log n), which is quicker than the naive technique, which would require O() comparisons to compare every pair of items.