Project 2 - Comparing Divide and Conquer with Brute Force Algorithms

Ying Zhang
CS 2223
11/21/2017

Question 1 What is the time efficiency for your brute force algorithm?

My Brute force solution compute the distance between each pair and return the smallest. The time efficiency is O(n^2)

$$T(n) = \sum_{i=0}^{n} \sum_{j=i+1}^{n} 1 = \sum_{i=0}^{n} (n-i-1) = n(n-1) = O(n^2)$$

Question 2 What is the time efficiency for your recursion algorithm?

Divide and Conquer strategy first find the *P[n/2]* middle point in the sorted array, then divide the given array in two halves. The first subarray contains points from P[0] to P[n/2],the second subarray contains points from P[n/2+1] to P[n-1].Next, recursively find the smallest distances in both subarrays. Let the distances be dl and dr. Find the minimum of dl and dr. Let the minimum be d. Let Time complexity be T(n). Sorting algorithm is O(nLogn). The above algorithm divides all points in two sets and recursively calls for two sets. After dividing, it finds the strip in O(n) time. Also, it takes O(n) time to divide the Py array around the mid vertical line. Finally finds the closest points in strip in O(n) time. So T(n) can expressed as follow:

T(n) = 2T(n/2) + O(n) + O(n) + O(n)
T(n) = 2T(n/2) + O(n)
T(n) = T(nLogn)