Name: Abu Nowshed Sakib

ID: 1705107

Complexity Analysis

I am analyzing runtime complexity step-by-step according to my implemented algorithm.

Step-1: Sorting the given points of houses by x and y coordinate separately. This takes O(n logn) time.

Step-2: Dividing the points with the middle line so that half of the points are on each side. Then partition the points in two sub-arrays according to y-coordinate sorted array. This partition takes c$\frac{n}{2}$ time. Complexity O(n).

Step-3: Then recursively finding the second closest pair of points in each sub-array takes T($\frac{n}{2}$) times. So, complexity is 2T($\frac{n}{2}$). Where T(n) is the total complexity of this problem.

Step-4: Now there can be such two points who are located in the different part of the middle line. To solve this we can consider a strip of width 2*d and height 2*d around a point (where d = second closest distance obtained in the step-3). There can't be more than seven points as such, we can prove that with geometry. This sub-problem will take O(n) time.

Step-5: Combining the sub-problem results will take O(n) time. Base case(when n<=3) will take O(1) time.

So, Finally we can write that:

Phase-1 (Sort):

Sorting : O(n logn)

Phase-2 (Divide and Conquer):

Partition : O(n)

Recursion : 2T($\frac{n}{2}$)

Strip-step : O(n)

Merge : O(n)

Base-Case : O(1)

So, T(n) = 2T($\frac{n}{2}$) + O(n) + O(n) + O(n) + O(1)

- ⇨ T(n) = 2T($\frac{n}{2}$) + O(n)

Using Master Theorem:

Comparing above recurrence relation with T(n) = aT(n/b) + f(n) we get-

a = 2, b=2, f(n)=O(n)

So, $n^{\log_b a}$ = $n^{\log_2 2}$ = n^1 = n = f(n) -> which falls in case-2 of master Theorem.

So, T(n) = O(f(n)* logn) = O(nlogn)

Therefore, Time Complexity of this problem is O(n logn).