

Lecture 5: January 18, 2018

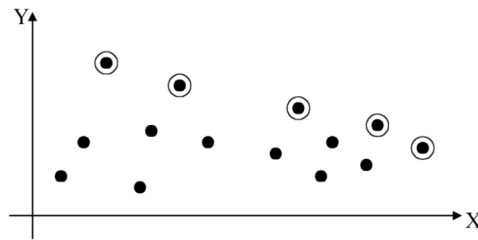
Lecturer: Bin Ma

Notes By: Harsh Mistry

5.1 Divide and Conquer Continued

Example 5.1 2D Maxima

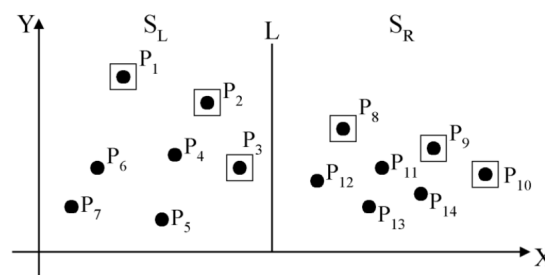
Let $(x_i, y_i), i = 1, \dots, n$ be n points on a two dimensional plane. A point (x_i, y_i) dominates (x_j, y_j) if $(x_i > x_j)$ and $y_i > y_j$. A point (x_i, y_i) is a 2D maximum if no other points dominate it. That is, for any $j \neq i$, either $x_j \leq x_i$, either $x_j \leq x_i$ or $y_j \leq y_i$



Taken from Bin Ma's Lecture Notes

if X and Y are treated as two different properties about a product. Then a maximum can provide some unique value proposition. However, a product that is dominated by some other products is hopeless.

The problem is to find all maxima from n given points. We solve this by divide and conquer.



Taken from Bin Ma's Lecture Notes

For simplicity, let's assume points have different x -axis. (More general cases to be discussed later). Divide the points into two halves.

Suppose the recursion returns the maxima from left and right halves, respectively. How do we construct the maxima of the whole point set?

First we notice that a maximum for the whole set must be a maximum of either the left or the right halves. Secondly, the right maxima are already maxima of the whole set. Thirdly, a maximum of the left half is a maximum of the whole set if and only if it is higher than all points at the right side.

we this first sort all points to x -axis. This will take $O(n \log n)$ time. Then the following algorithm will work.

Algorithm Maxima-DC

Input : a list of points $(x_1, y_1), \dots, (x_n, y_n)$ sorted across to x -axis

1. If $n < 5$ solve the problem by definition and return the maxima
2. Solve the first $n/2$ and last $n/2$ points recursively. Denote solutions be C_1 and C_2 , respectively
3. Compute the largest y -axis in the last $n/2$ points. Denote it m_y
4. Return $C_2 \cup \{(x, y) \in C_1 \mid y \geq m_y\}$

The correctness follows from the above discussion. Detailed proof is left as an exercise

Time complexity: $T(n) = 2T(n/2) + n$ From Master theorem, $T(n) = O(n \log n)$

Now consider the slightly more general case where two points may have the same x -axis. We can reuse the same algorithm. The only difference is that when ordering two points with the same x -axis, we put the one with smaller y axis at left. One can verify that the same algorithm and same proof