Maximal points

Given a set $S$ of $n$ tuples, find an
the maximum subset $V \subseteq S$ s.t.
no tuple in $V$ is dominated by any
other tuple. A tuple: $(x_1, x_2, x_3, \ldots, x_d)$

$O(n^2)$ brute force algorithm is
easy: Just do all pairwise
comparisons to eliminate the points that
are not maximal.
Algorithm 1

Step 1: Sort the points on the basis of their $x$ coordinates $O(n \log n)$

Step 2: Scan the points in reverse order of their $x$ coordinates

keep track of the highest $y$ coordinates among the points scanned, say $Y_{\text{max}}$

Step 3: If the present point $(x_i, y_i)$ is such that $y_i < Y_{\text{max}}$ then $(i)$ is maximal

else $(x_i, y_i)$ is maximal and

$Y_{\text{max}} \leftarrow y_i$

until all points are scanned

Proof that the algorithm correctly identifies all the maximal points
By induction on the loop (loop invariants)

All the points that have been scanned have been correctly classified

Base case: $n = 2$ coordinate

Time and Space Complexity Analysis

$O(n \log n) + O(n) \rightarrow O(n \log n)$

Sorting

Sweep or Line sweep

Try Divide-and-conquer

Divide by partitioning

Merging is easy $O(n)$

Arbitrary divide with two sets

Merging is more complex
\[ T(n) = 2T(n/2) + O(n) \]
\[ \Rightarrow T(n) = O(n \log n) \]

Consider the following input

By filling out all the points dominated by the latest maximal point

\[ T(n) = \frac{1}{n} n \]

\# maximal points
Output-size sensitive algorithm

Better output-size sensitive algorithm have running time $O(n \log n)$.