IIT Kanpur

CS345A - AlgorithmsII

Assignment 1

Anjani Kumar   Sumedh Masulkar

11101   11736

January 12, 2014

0.1 Non Dominated Points

0.1.1 Θ(n log n) algorithm for non-dominated points in a plane.

Overview of algorithm Given set of points P in a plane.
Divide Step:

1. If there is only one point in a plane, the point itself is non-dominated set of points. Hence, return P.

2. If there are two points, if any point has both x and y co-ordinates both greater than that of other point, then return that point, else return P.

3. Else find the x-median of the points and divide the plane into left half plane and right half plane using the median. And now call the function for both the half planes.

Conquer Step:
Goal: Given the non-dominated points of the two half planes, merge the solution of smaller parts to get the solution of the bigger plane.

Assuming we have two sets of points P_1 and P_2, where P_1 is the set of non-dominated points of the left plane, and P_2 is the set of non-dominated points of the right plane respectively.

The x-coordinate of all the points in right plane are obviously greater than x-coordinate of all the points in the left plane. Thus, we only need to eliminate points from P_1 that are dominated by points in P_2.
Since x-coordinate of points in P_2 is always greater, we only need to look for the y-coordinates.
Let y be the point in P_2 with maximum y-coordinate. Then the dominated points in P_1 are all the points whose y-coordinates are less than y-coordinate of y.
Thus the solution of the plane will be { P_1 - {points in P_1 with y-coordinate < y } } $\cup P_2$.

Pseudo-Code.

```
NonDominatedPts(set of points P){
        //Returns set of non dominated points from P.
if |P|==1 then
 |  return P;
else if |P|==2 then
 |   let p₁ and p₂ be the two points in P;
 |   if x₁ > x₂ and y₁ > y₂ then
 |    |  return {p₁};      //p₁=(x₁,y₁), and p₂=(x₂,y₂)
 |   else if x₁ < x₂ and y₁ < y₂ then
 |    |   return {p₂};
 |   else
 |    |  return P;
 |
else
 |   p* ← x-median(P);
 |   (L,R) ← split(P, p*);
 |   P₁ ← NonDominatedPts(L);
 |   P₂ ← NonDominatedPts(R);
 |   P₁ ← P₁ sorted along y-axis;
 |   y ← max y-coordinate in points of P₂;
 |   P₁ ← P₁ - {all points in P₁ whose y-coordinate ≤ y} ;
 |   return (P₁ ∪ P₂) ;
}
```
Algorithm 1: O(nlogn) algorithm to find Non Dominated Points

0.1.2 Θ(n log h) algorithm for non-dominated points in a plane.

Explanation

Pseudo-Code.

```
NonDominatedPts(set of points P){
        //Returns set of non dominated points from P.
```
if $|P|==1$ **then**
| return P;
else if $|P|==2$ **then**
| let p_1 and p_2 be the two points in P;
| **if** $x_1 > x_2$ *and* $y_1 > y_2$ **then**
| | return $\{p_1\}$;   //p_1=(x_1,y_1), and p_2=(x_2,y_2)
| **else if** $x_1 < x_2$ *and* $y_1 < y_2$ **then**
| | return $\{p_2\}$;
| **else**
| | return P;

else
| p* $\leftarrow$ x-median(P);
| (L,R) $\leftarrow$ split(P, p*);
| $P_1 \leftarrow$ NonDominatedPts(L);
| $P_2 \leftarrow$ NonDominatedPts(R);
| $P_1 \leftarrow P_1$ sorted along y-axis;
| y $\leftarrow$ max y-coordinate in points of P_2;
| $P_1 \leftarrow P_1$ - {all points in P_1 whose y-coordinate $\leq$ y} ;
| return $(P_1 \cup P_2)$;
```
}
```
Algorithm 2: O(nlogh) algorithm to find Non Dominated Points