Department of Computer Science
Ashoka University

Design and Analysis of Algorithms: CS-3210-1

Programming Assignment 1 - Proof of Correctness

Name: Dhruman Gupta

Here, I will be detailing the proof of correctness of the algorithm to show that it indeed computes all the maximal layers correctly. The algorithm takes a set of n points, and outputs $k \leq n$ layers, where the $1 \leq i \leq k$ layer is the i^{th} maximal layer.

The Algorithm

Note: for simpliclity, we will assume that for each layer i, l_i represents the maximum y value of all elements in layer i.

1. Sort the n points in descending order on the x-axis. If two points share the same x-coordinate, then sort descending on y-axis.
2. Initialize an empty list L. Each element in this list will represent a maximal layer. Also, $L[i]$ will represent the i^{th} maximal layer. Each layer is represented by a list, too, sorted descending on the x-axis. Note, that given an index, we can access the layer in $O(1)$ time. So, accessing the greatest y-coordinate in a layer can be done in $O(1)$ time.
3. Begin sweeping from the right - i.e, the first point of the sorted array.
4. At each element $i = (x_i, y_i)$, we want to assign i to the correct layer. Do a binary search on the highest y-coordinate of each layer in L, to find the greatest layer j such that $y_i > m_j$.
   (a) If such a layer exists, assign i to layer j.
   (b) If no such layer exists, then create a new layer and assign i to it. Add this to the end of L.
5. When all points have been assigned to a layer, return L.

This algorithm correctly computes the maximal layers. We will now prove this.

Proof of Correctness

Below are a set of claims that will then be used to prove the correctness of the algorithm.

Claim: $i < j \implies l_i \geq l_j$ - i.e, L is sorted in descending order of l_i. This will also be an invariant.
Proof:
Initialization: The claim is vacously true at the beginning of the algorithm, as L is empty.

Maintenance: Assume that the claim is true at some step $i - 1 \geq 0$. We will show that it is true at step i. We are inserting the $i^{th} = (x_i, y_i)$ element in the sorted set. The insertion step finds:

$$j = \underset{k, \text{ s.t } y_i > l_k}{\arg\max} \; l_k$$

Now, we have two cases: such a j exists, or not.

1. j **exists.** Then, $y_i > l_j$, also $y_i \leq l_{j-1}$, because if it does not, then $\arg\max$ would have chosen $j - 1$. So, after y_i is added to layer j, l_j will be updated to y_i. Also, l_{j-1} will remain the same. So, the claim is true.
2. j **does not exist.** Then, a new layer is created, and y_i is added to it. The new layer will have $l_{k+1} = y_i$. We know that $\forall j, y_i \leq l_j$. So, this implies that $l_{k+1} \geq l_j$ for all j. So, the claim is true.

So, L is indeed sorted in descending order of l_i.

Claim: The algorithm computes all the maximal layers.
Consider the following invariants:

1. L is sorted in descending order of l_i. (proven)
2. At step i, $\forall k, L[k]$ has all the elements in k^{th} maximal layer from the first i elements of the sorted array.

Trivially, if these invariants hold, then the final output L is correct. I will prove the second invariant.

Proof:

Need to show the initialization, maintainance, and termination steps.

Initialization: The claim is vacously true at the beginning of the algorithm, as $i = 0$. L is empty.

Maintenance: Assume that the claim is true at some step $i - 1 \geq 0$. We will show that it is true at step i. We are inserting the $i^{th} = (x_i, y_i)$ element in the sorted set. Say we insert point i into the j^{th} layer. This insertion would be correct if and only if:

1. The point i dominates no point in layer j, and is dominated by at least 1 point in layer $j-1$ (if $j \neq 1$), and that no point in layer $j+1$ dominates i (if $j \neq k$ where k are the total number of maximal layers).
2. For all future insertions, the above holds for this point.

As we are sweeping in descending order, we know that of all the points in all built-up layers, point i has the lowest x value. If we insert into layer j, it means that $l_{j-1} \geq y_i$. This implies that that particular point in layer $j-1$ dominates point i. Also, it means that $l_{j+1} < y_i$. So no point in the layer $j+1$ has a y value more than i. Thus, no point in the next layer dominates i.

This concludes the proof that it is correct.

(P.S.) I know this proof is not complete and wrong in some places (need to show that the termination implies maximal layers), due to time restrictions I have omitted to write that.