A Reseach Project Report (Midsem)

On

Reporting and counting maximal points in a query orthogonal rectangle

By

Ananda Swarup Das, Prosenjit Gupta, Kishore Kothapalli, Kannan Srinathan

PUBLISHER: Journal of Discrete Algorithms

DAS ANAMIKA ABHOYPAD

2014H313059H

Under the supervision of

TATHAGATA RAY

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

HYDERABAD CAMPUS

ABSTRACT

In this work we show that given a set *S*of *n* points with coordinates on an *nxn* grid, we construct data structures for reporting and counting the maximal points in an axes-parallel query rectangle in sub-logarithmic time. We assume our model of computation to be the word RAM with size of each word being log*n* bits.

INTRODUCTION

Database system plays a major role in the industries as well as in our daily life. Hence, optimisation and efficiency are the important factors of the database system.

The given research paper focuses on database community problems. The domain relates to “skyline points”.

Lets walkthrough with an example,

Suppose, you have a 10 days suspension from the mundane work and you have decided to go on a holiday. The favorite scenario that you prefer, is a beautiful sunset on a beach. You open some travel site and search for a hotel that is near to the beach! Plus the expense must be relative to your pocket size.

There are three possible serach results:

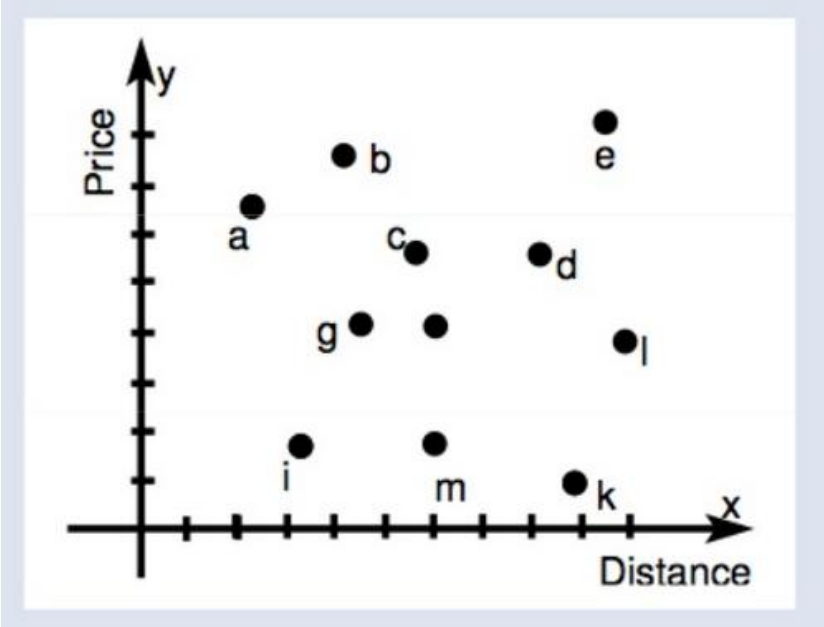
1. Cheapest hotel but not close to the beach.

2. Closest to the beach but expensive hotel.

3. Neither close to the beach nor expensive hotel , relatively fits the criteria of your choice.

Lets take a look at the graph with the datasets that contains the details of the distance to the beach and the price of the hotel:

http://www.slideshare.net/victormier/skyline-queries



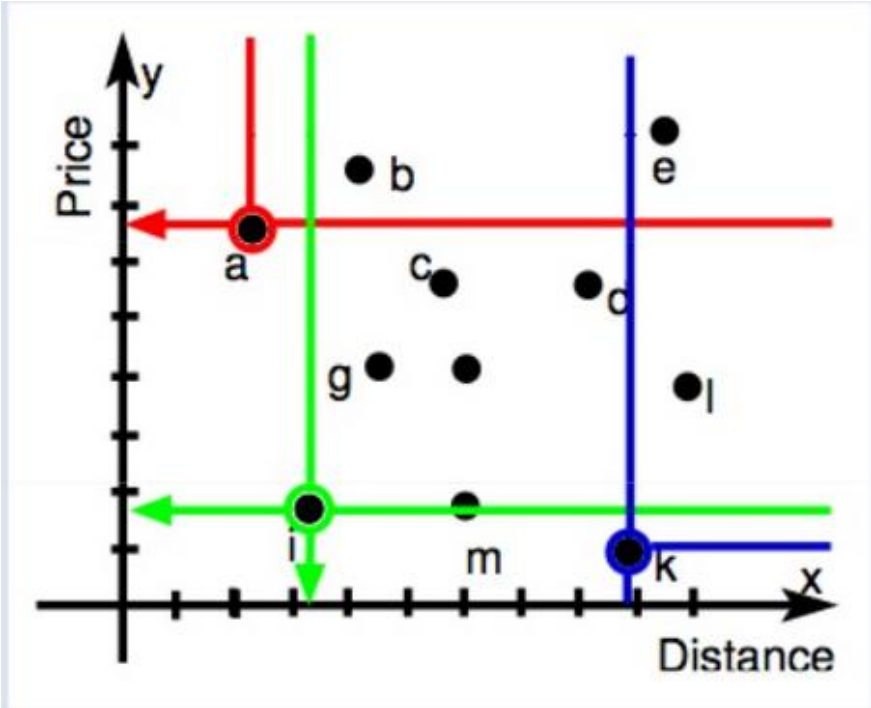
x-axis= Distance, y-axis= Price

The interesting points are a, i, k. Why??

We can see that point a has the least distance but high cost, point k has the lowest price with the not so close to the beach.

i neither has shortest distance nor lowest price. i has distance less than k and price more than a. i is not **dominated** by either of them

All the other points are dominated by the points a, i and k.



Hence the points that are not dominated by another points are called as skyline points. The skyline points for the above graph are a, i, k.

The dataset: (price , distance).

The skyline points are important to the large datsets that are stored in the database.

DEFINITION: Skyline is a set of tuples of information that are important or of special interests to us.

A point dominates another point if it is good in all dimensions or atleast in one dimension.

A point p(x1, y1) is dominated by point q(x2, y2) if x1<y1 and x2<y2 in terms of speciality.

The research paper has described **skyline points as maximal points**.

The reseach paper deals with 2 Dimensional skyline points i.e., the co-ordinates of x-axis and y-axis.

PROBLEM DEFINITION

The user is not concerned with the entire dataset. A user enters his/her queries i.e. window size and accordingly the result is displayed. The simple appraoch goes as to check every dataset with the given queries and give the output. If the datasets are too large, the time taken for the search queries increases and hence expensive in terms of query time.

The solution uses a data structure of size $O(n \log n)$and can be queried to report maximal points in $O(\log 2n + k)$time, where k is the size of the output.

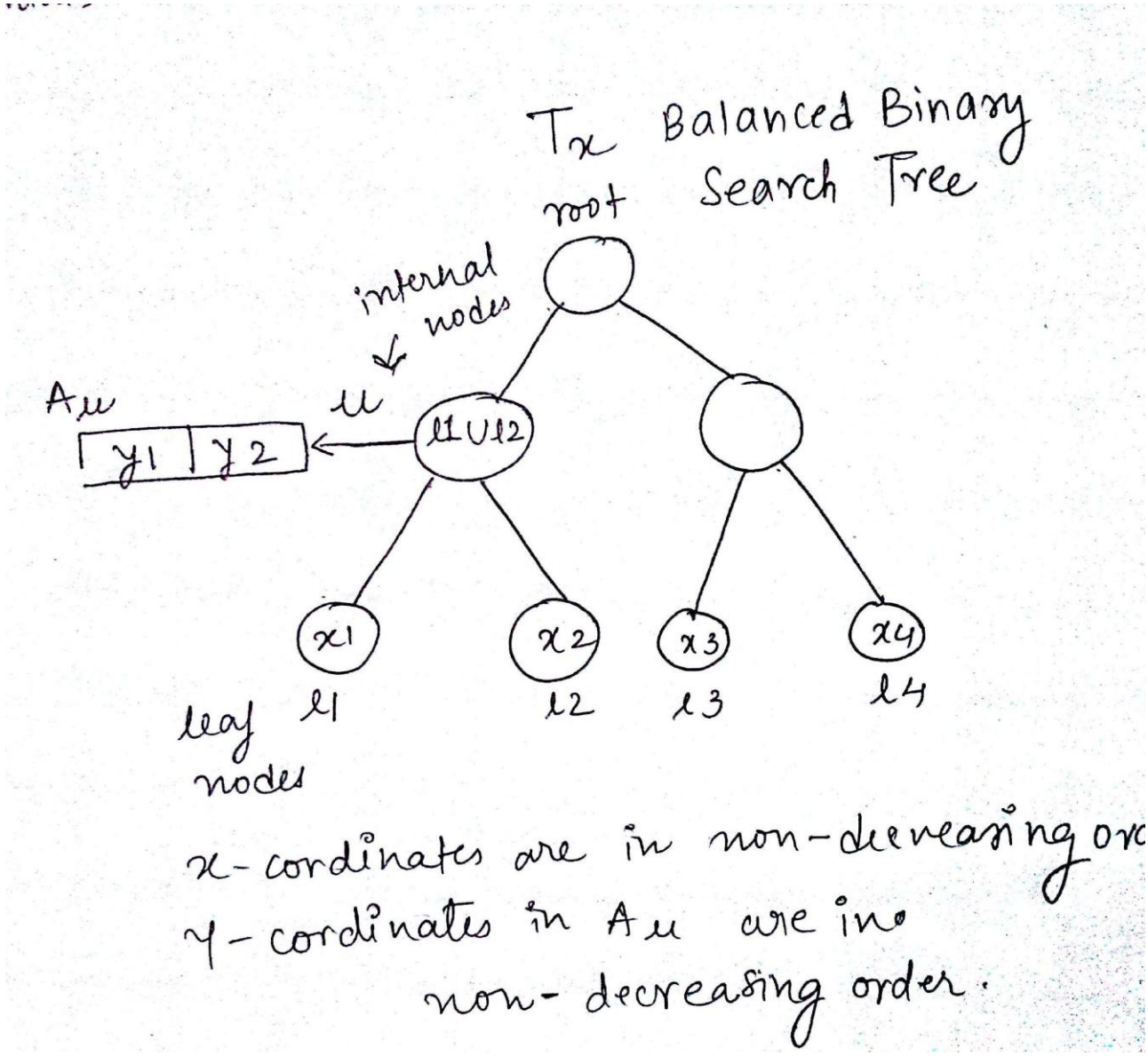
A proposed solution for the word RAM model of computation with size of each word being $\theta(\log n)$and propose a data structure of size $O(n*(\log n/\text{loglog}n))$that can be queried in $O((\log n/\text{loglog}n)+k)$time.

The maximal point also reduces the size of the result. We have an algorithm that computes this size. It is useful in the context, there is a need to transfer the result using network bandwidth. We propose a data structure of size $O((n*(\log(\text{^}3)n/\text{loglog}n))$that can be queried in $O(\log n/\text{loglog}n)$time.

RANGE TREE WITH FRACTIONAL CASCADING

We present a data structure with $O(n \log n)$space and $O(\log n + k)$query time i.e., 2-dimensional range tree (with fractional cascading pointers) in order to support "2-dimensional range maximal points" reporting queries.

Given a Set S of points (dataset) with x and y cordinates and a query rectangle q (user queries). We can process the points that are in S ∩ q;



We have four points (x1, y1), (x2, y2), (x3, y3) and (x4, y4).

We Build a tree Tx that is balanced binary tree that has leaf nodes l1, l2, l3, l4.

We store the x co-ordinates in non- decreasing order of their values at the leaf nodes i.e., x1<x2<x3<x4

Let μ be an internal node which is union of the discrete intervals associated with the leaf nodes of the subtree rooted at μ.

Each internal node μ also maintains an auxiliary array *A*μ which stores the *y*-coordinates of the leaf nodes present in the subtree rooted at μ. These *y*-coordinates are sorted in non-decreasing order of their values i.e., y1<y2

Leaf nodes of subtree rooted at μ are l1 and l2.

For any node φ belongs to Tx, itmust be noted that int(φ) =int(u) ∪int(w)where u, ware respectively the left and the right children for φ. Also Aφ=Au∪Aw. Thus, with each element y belongs to Aφ, westore two pointers pointing to the smallest values greater than yin Auand Awrespectively. The idea of maintaining such pointers in known as fractional cascading.

The storage space needed by the range tree is O(n logn).