Computational Geometry CSL 852

Lecture 26

Topic : Range Searching

Given a set S of n points in d dimensions, we are allowed to build an appropriate data structure that supports query of the following kind

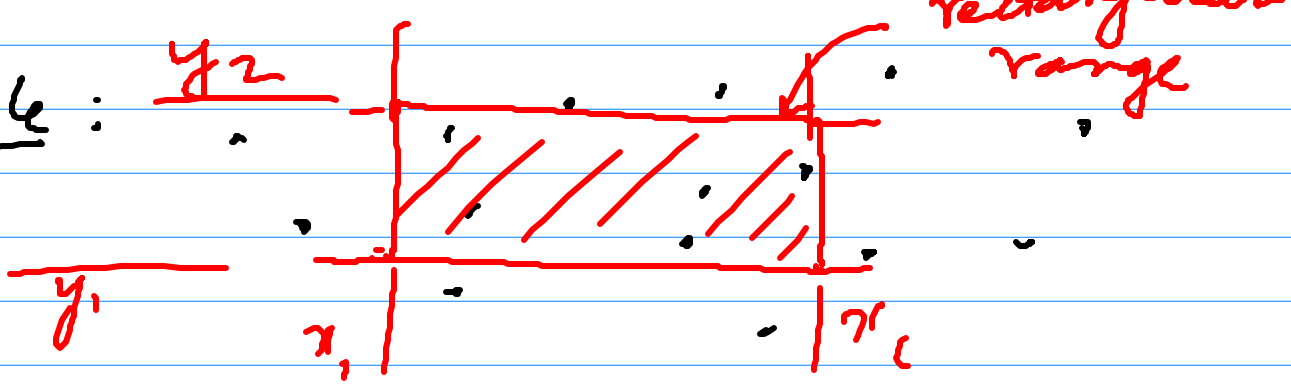
- For some query "range" $\triangle$, we should return

(i) All points in $\triangle \cap S$ (report)

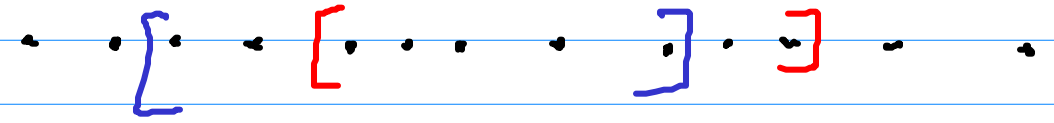
(ii) The number of points $|\triangle \cap S|$ (counting)

"Range" is a predefined family of subsets of E^d.

Example :



A general data base query can be posed as a "hyperrectangle" range query in $\mathbb{E}^d$.



Suppose there would be only 'I' query.

Trivial approach: For every $p \in S$, determine if $p \in \Delta$?

Usually $O(1)$ test

$O(n)$ trivial algorithm

For multiple queries, it makes sense to build a data structure so that the overheads of construction is justified and we can return the answers quickly.

Size of d.s.

reporting: $O(\text{polylog} + \text{output})$

counting: $O(\text{polylog})$

Range searching problems

Static case

dynamic case
(where the underlying point set can change)
- Time to update the data structure

Ideally: polylog