Data Structures for Computational Geometry

Mashhood Ishaque

21st June, 2007

Computational Geometry Research Group

http://www.cs.tufts.edu/research/geometry/

Data Structures

Preprocess data to support:

1. Fast queries
2. Fast updates
3. Using little space

Data Structures

- There are two different approaches to answer a query.
- Algorithmic (one-shot) approach:
  – Solve the problem from scratch for every query.
- Data structural approach:
  – Spend some time preprocessing the data.
  – Answer queries quickly.

Data Structures

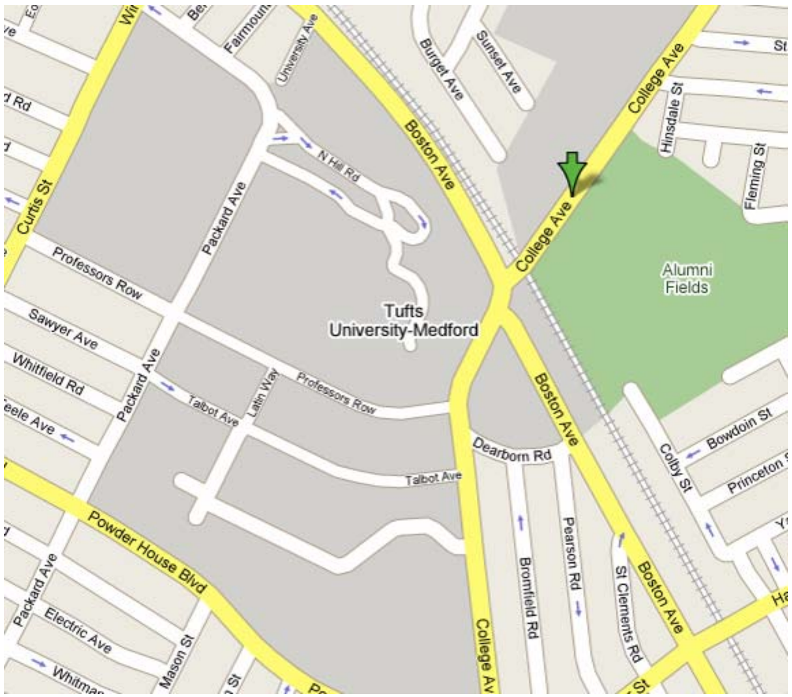
- Algorithmic solution is efficient if we have to answer only once.
- But if we are going to query many times data structural approach is preferable.
- We measure efficiency in term of computational complexity.

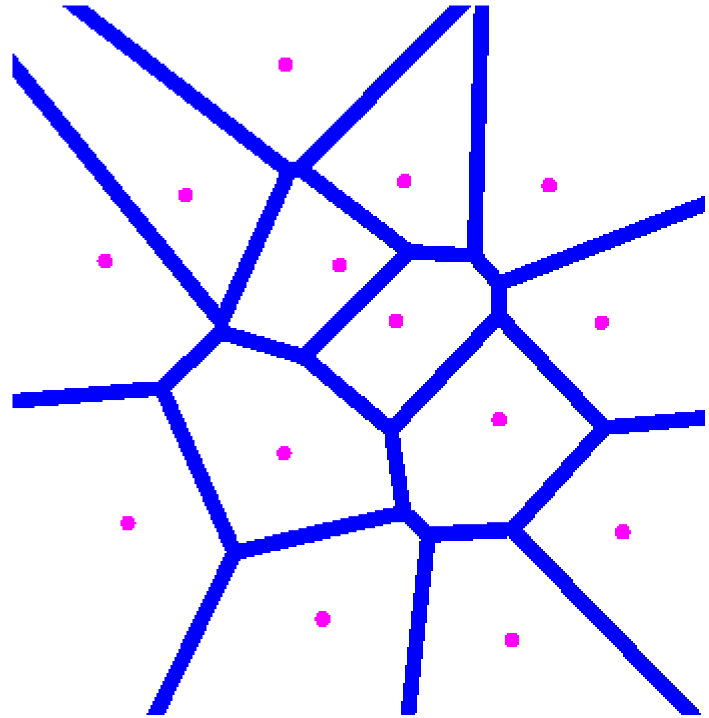
Dynamic Data Structures

- Sometime we need to insert new data or delete some old data. Insertions and deletions are known as updates.
- A data structure that supports updates is called dynamic.

Five Problems in Geometry

Post Office Problem (Voronoi Diagrams)

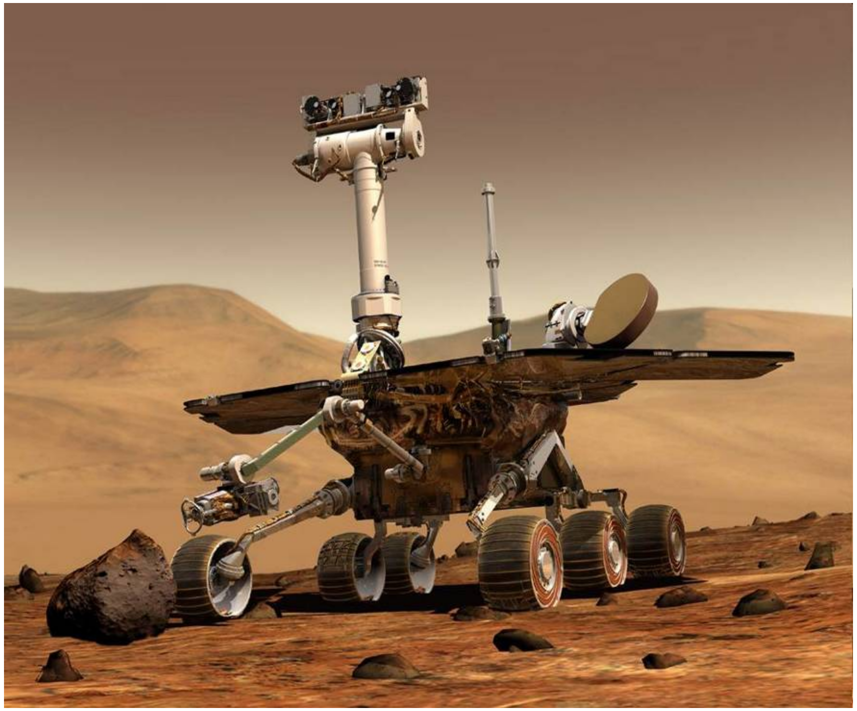


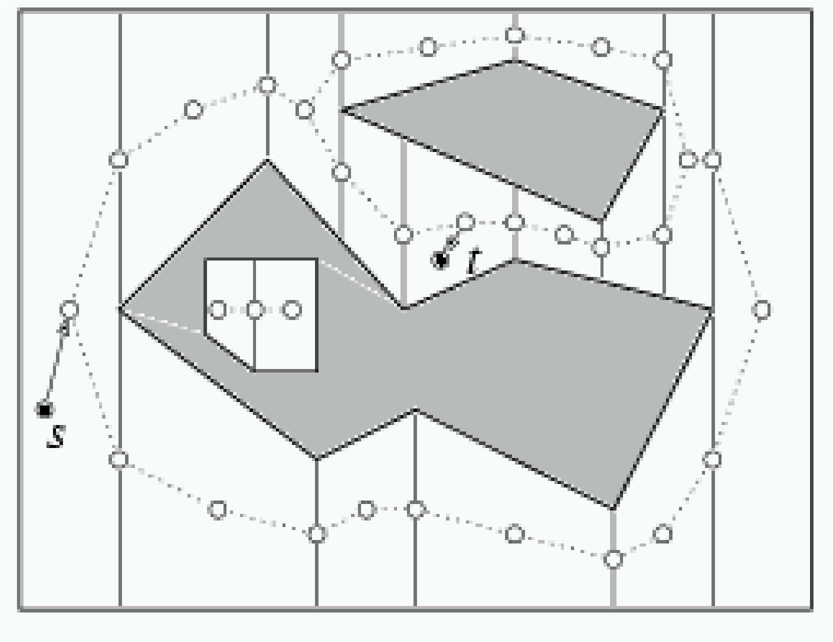


Where is the closest post office (pizza place)?

Five Problems in Geometry

Robot Motion Planning (Trapezoidal Maps)

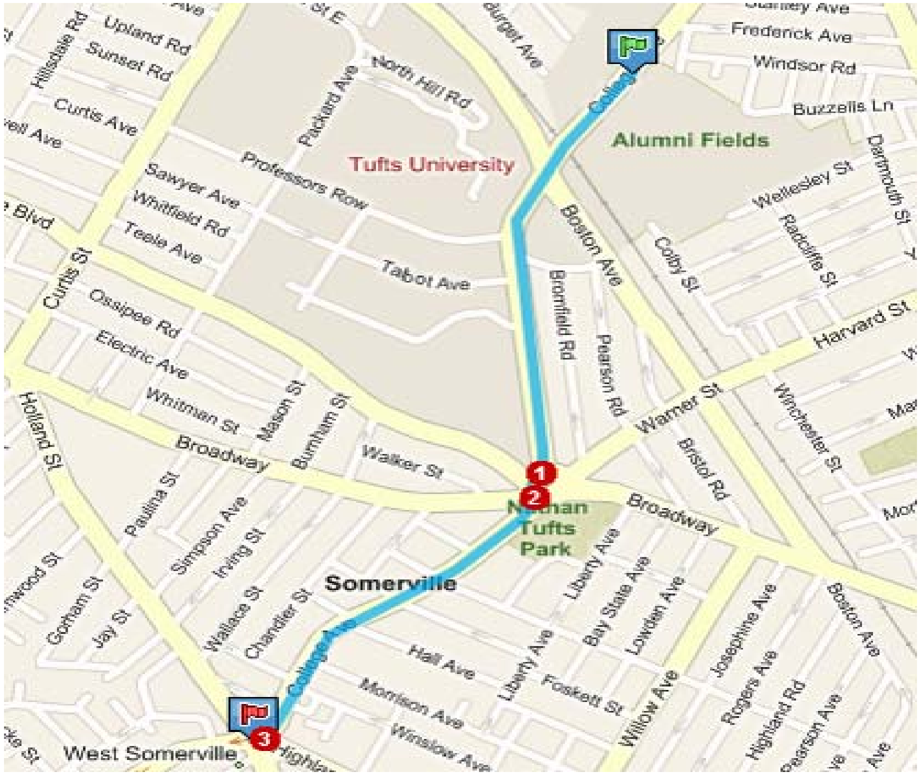


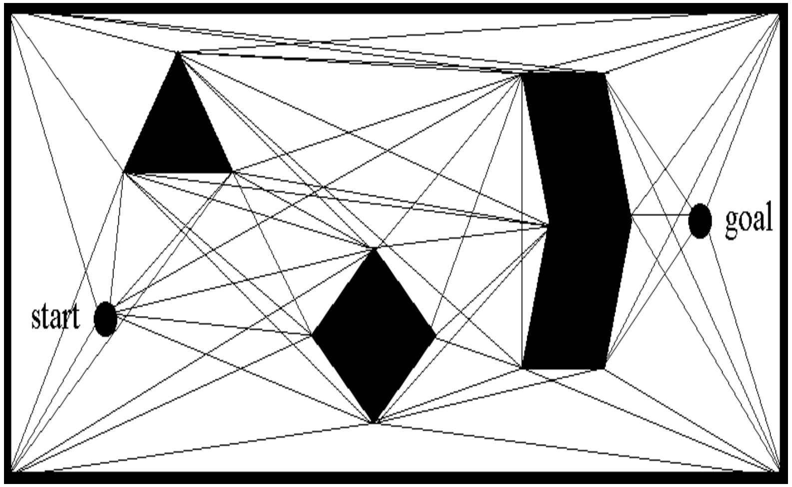


Find an obstacle-avoiding path from source to destination.

Five Problems in Geometry

Finding the Shortest Route (Visibility Graphs)

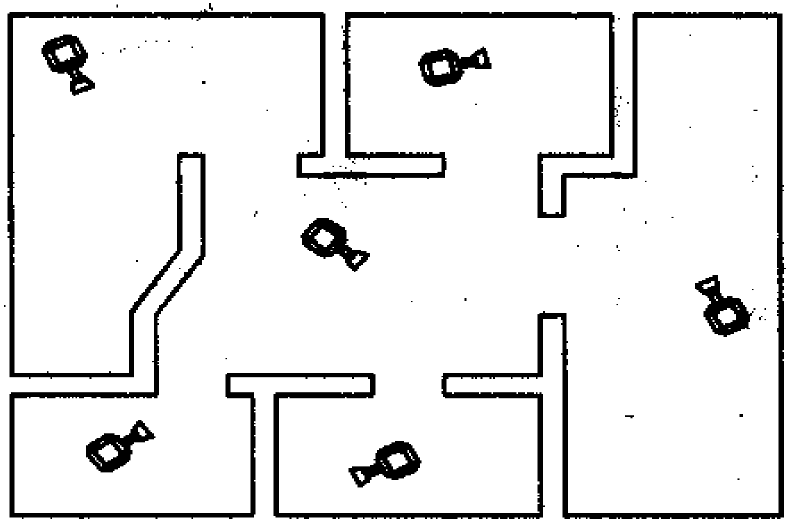




Find the shortest (distance/time) path from Halligan to Davis Square.

Five Problems in Geometry

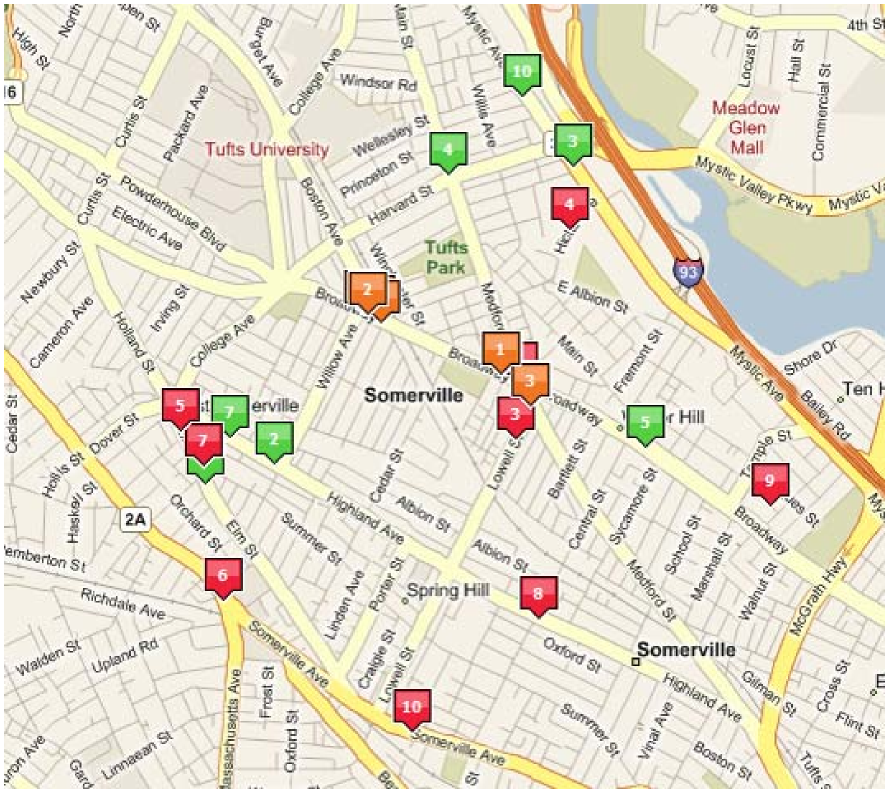
Guarding an Art Gallery (Triangulation)

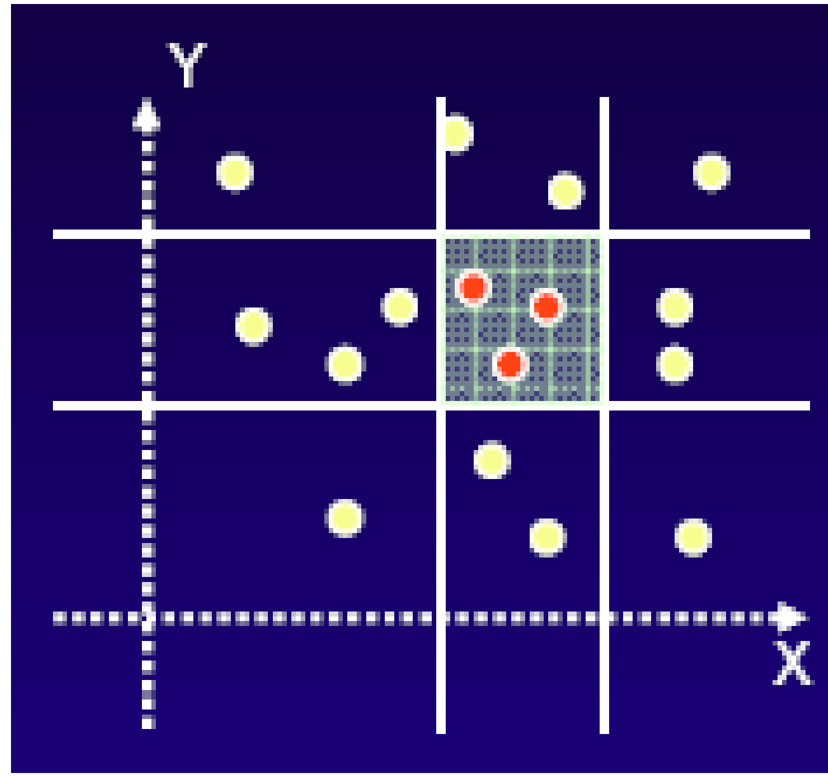


1. How many cameras do we need to guard a given gallery?
2. Where should we place these cameras?

Five Problems in Geometry

Querying a Database (Range Searching)



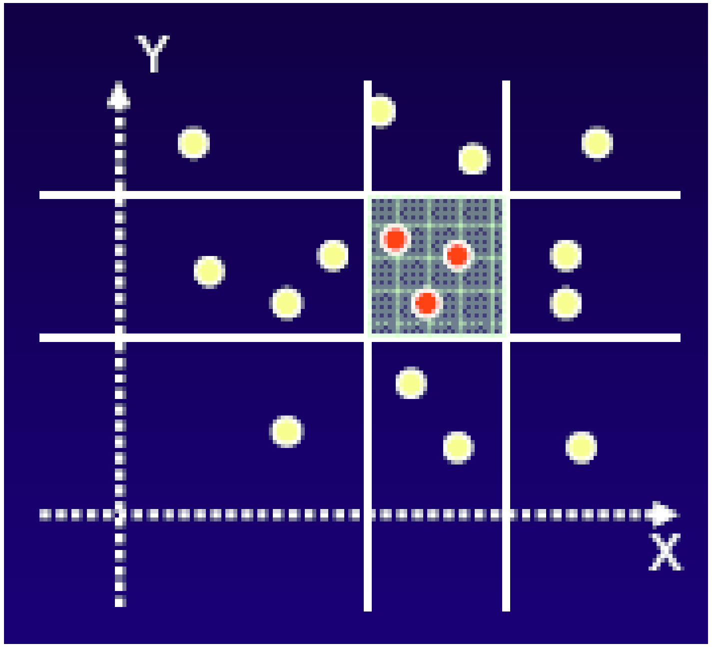


How many restaurants are there around Tufts University?

Range Searching in Detail

Orthogonal Range Searching

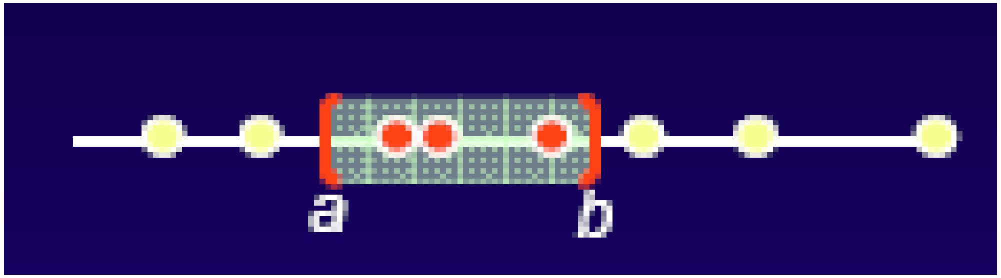
Given a set of n points in the plane, preprocess them such that reporting or counting the points inside query (axis-parallel) rectangle would be most efficient.



Range Searching in Detail

Range Searching—1D

1. Points are real numbers
2. Query is just an interval [a,b]



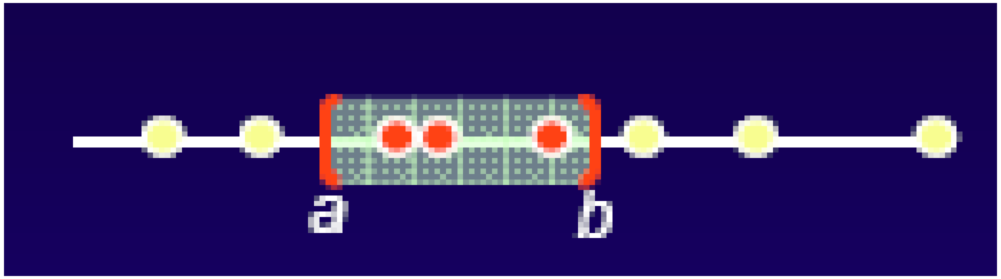
What is the simplest (naïve) solution?

Just look at each point, and count those points which lie within the interval.

Range Searching in Detail

Range Searching—1D

1. Points are real numbers
2. Query is just an interval [a,b]



What is a more efficient solution?

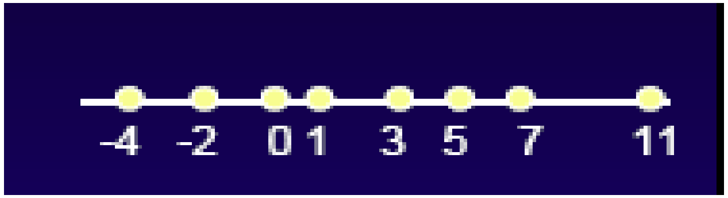
Sort the points (preprocessing), and then for each query just use binary search.

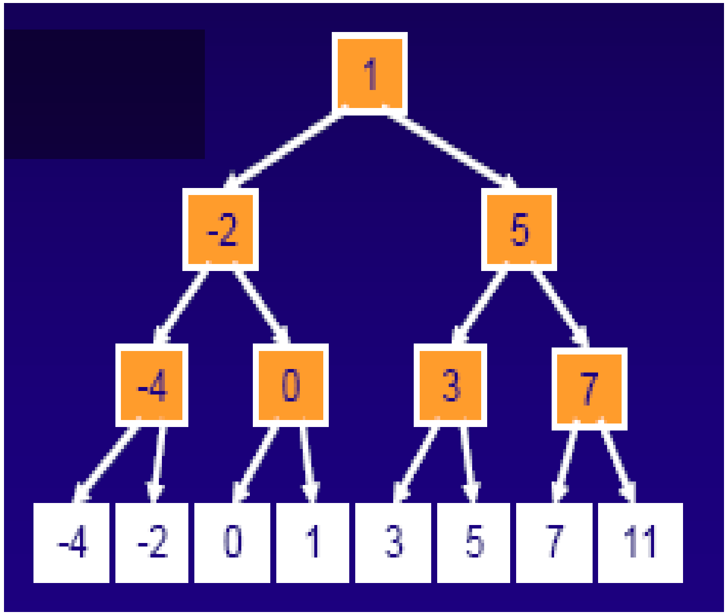
Balanced Binary Search Trees

Range Searching—1D

We can store the sorted points in an array. But arrays do not support efficient insertion/deletions.

So we need balanced binary search trees for dynamic range searching.

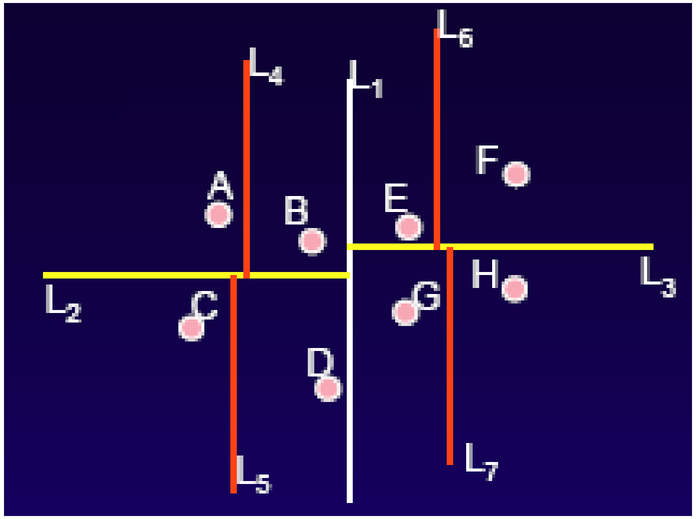


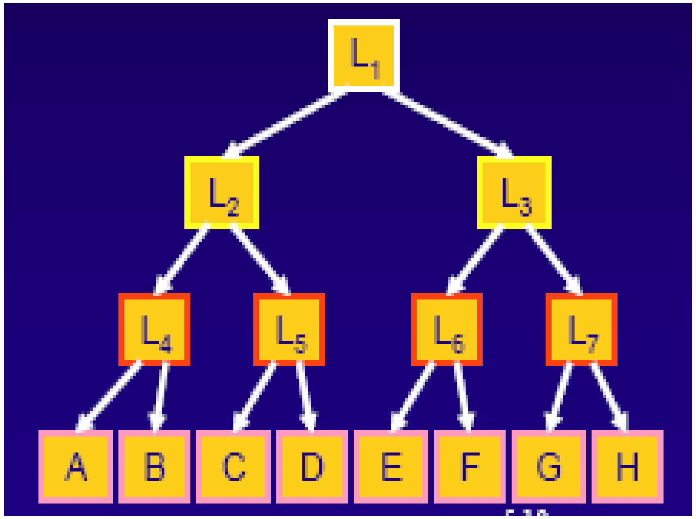


Range Searching in Detail

Range Searching—2D

1. Given n points in the plane
2. Query is an axis-parallel rectangle





Solution = kd Trees

Thanks for Listening!

Related Courses

1. Data Structures (Comp 15)
2. Algorithms (Comp 160)
3. Computational Geometry (Comp 163)