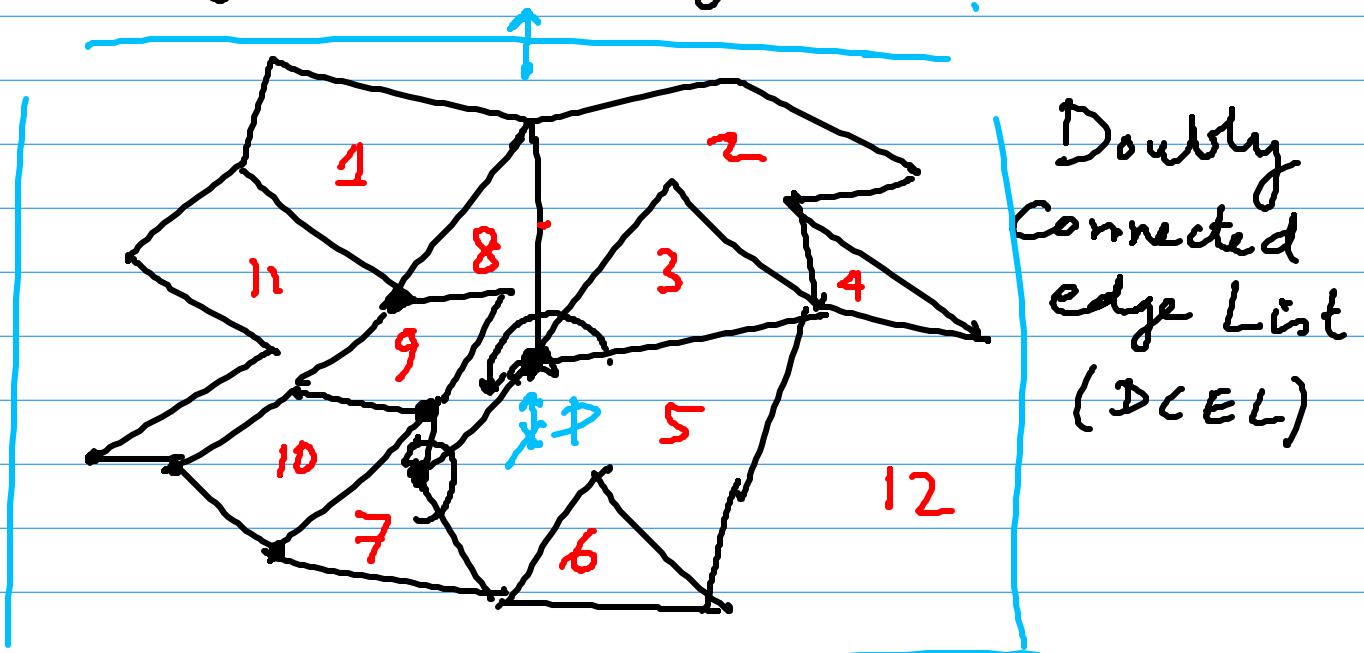


Computational Geometry Lecture 14

Topic : Point Location & Triangulation

Given a planar subdivision (straight lines only)



Build a data structure that supports queries of the following kind

"For a point $p = (x', y')$, which region contains p ."

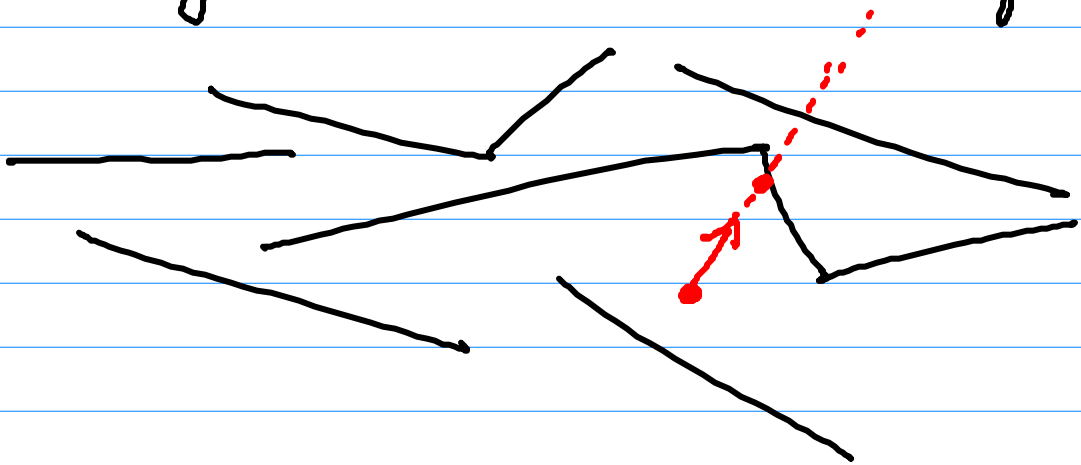
Parameterized by # vertices, # regions

Problem size corresponds to representation of this planar subdivision

Ray shooting

Given a set of edges (say non intersecting)
build a data structure to support
queries of the following type

vector " If we shoot a ray, which
segment does it hit first

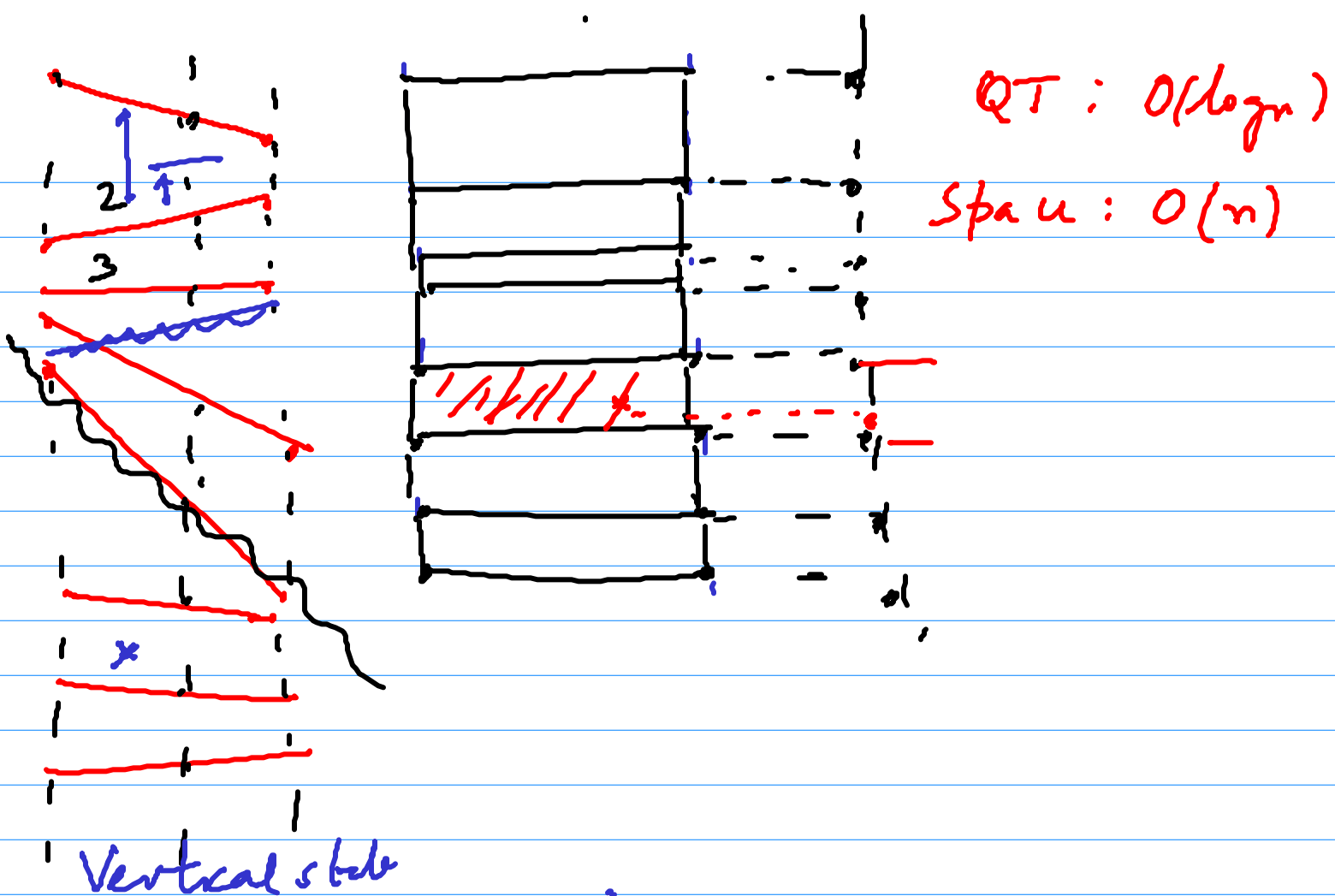


In the dynamic version, segments
can be inserted / deleted.

Our goal is to obtain

- (1) linear space data structure
- (2) logarithmic query time
- (3) preprocessing (construction) time

Space / query time tradeoff



Arbitrary Planar

Subdivisions can be partitioned into vertical slabs, such that two binary searches suffice for planar point location. (vertical ray shooting)

However space can be $\Omega(n^2)$.

(Suppose no two vertices lie on the same vertical line)

"Persistent data structure" for storing similar lists

