Problem: Point Location

given planar subdivision with $n$ vertices
build data structure s.t.
given query pt $q$, find region containing $q$
equiv: find line segment immediately above $q$

by Euler's formula, 
# edges/ faces = $O(n)$

App.: nearest neighbor search in 2D

Method 0:
divide into n vertical slabs
store y-sorted list in each slab

\( \Rightarrow \) query time \( O(\log n + \log n) \)
\( \text{binary search in } x \)
\( \text{binary search in } y \)

Space \( O(n^2) \)
preproc time \( O(n^2 \log n) \)

Method 1: Segment Tree

given \( n \) disjoint line segments intersecting slab \( \sigma \),

divide by median \( x \)
remove all long segs in \( \sigma \)
& store them in y-sorted list
recurse in left & right

**Def**
segment \( S \) is long \( \in \sigma \)
if it completely cuts across \( \sigma \).
Query algm, for query pt q:
find long seg immediately above q by binary search
if q left of median x
  recurse left
else recurse right
return lowest seg found.
Binary searches at $O(\log n)$ nodes along a path
$O(\log^2 n)$ Query time

How to speed up query?
Issue: parent list & child list not related...

**Method 2: Segment Tree + “Fractional Cascading”**
(Chazelle, Guibas ’86)

**Idea:** pass a $\frac{1}{b}$ fraction of parent list to child list

$b = 3$
Let $L(u)$ = original y-sorted list at node $u$
will generate an “augmented list” $L^+(u)$
Define $\text{sample}(L)$ = sublist of $L$ formed by
taking 1 out of every $b$ elements.
for each child $v$ of $u$,
let $L^+(v) = L(v) \cup \text{sample}(L^+(u))$
store succ ptrs between $L^+(v)$ and $\text{sample}(L^+(u))$