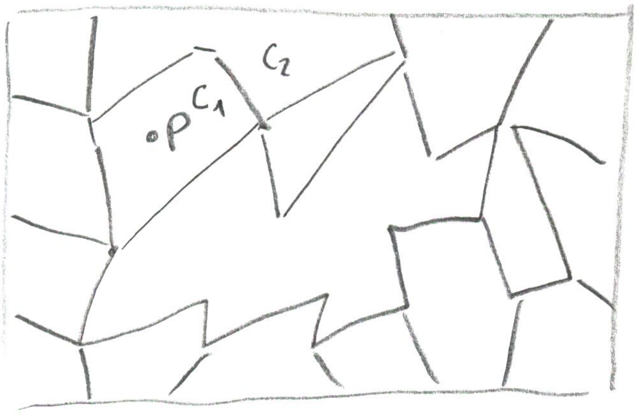
Point location

de Berg et al.
Computational Geometry

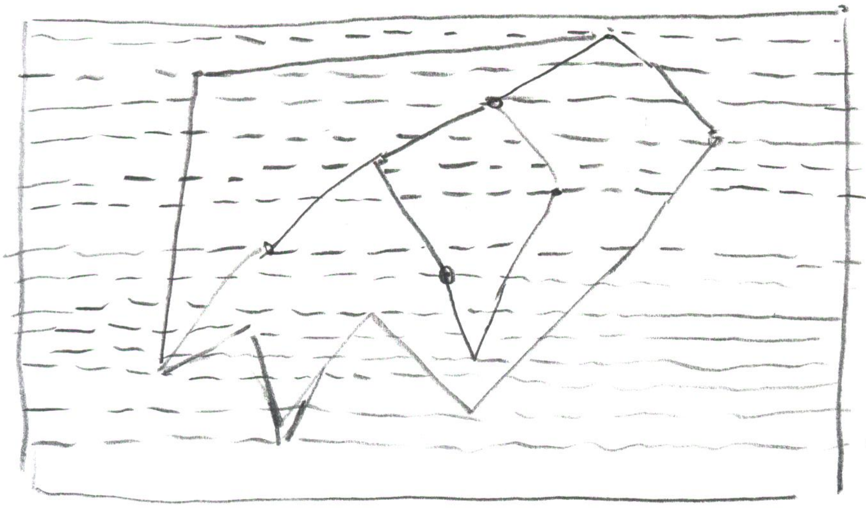
Cell arrangement: For example Voronoi Diagram

Decomposition into polygons

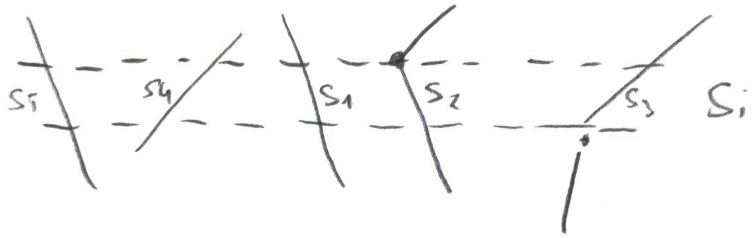


Query point p
Find cell C_i that contains p

Simple method: Strip-decomposition



- vertical / horizontal lines through any vertex
- Query structure for Y-coordinates
  binary tree T_Y
- Query structure for any strip S_i

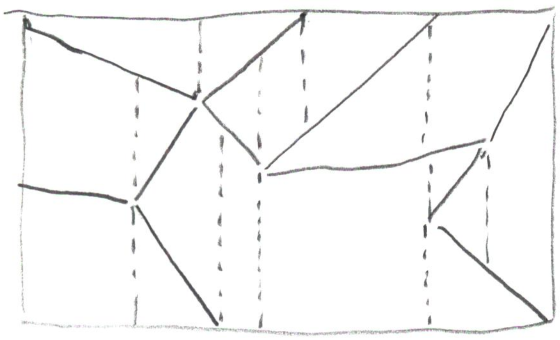


Regions subdivided by segments
binary tree T_X

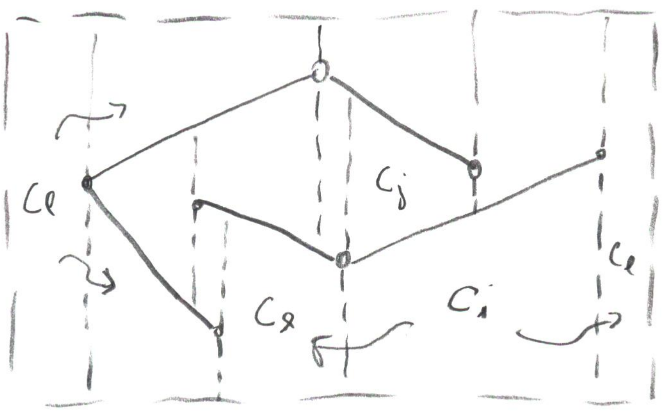
Point location: $2\times\ O(\log n)$

Problem: $\Theta(n^2)$ strips
long segments intersects many segments

Solution: Trapezoidal decomposition
Let the segments and vertical segments



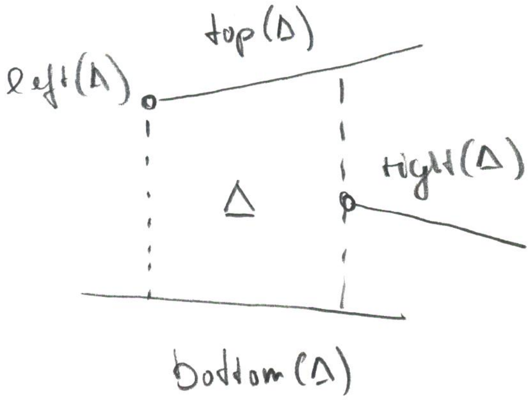
general: bounding box
n non-intersecting line segments (end points in common is allowed)
(general position, non-equal endpoints
i.e. different coordinates)



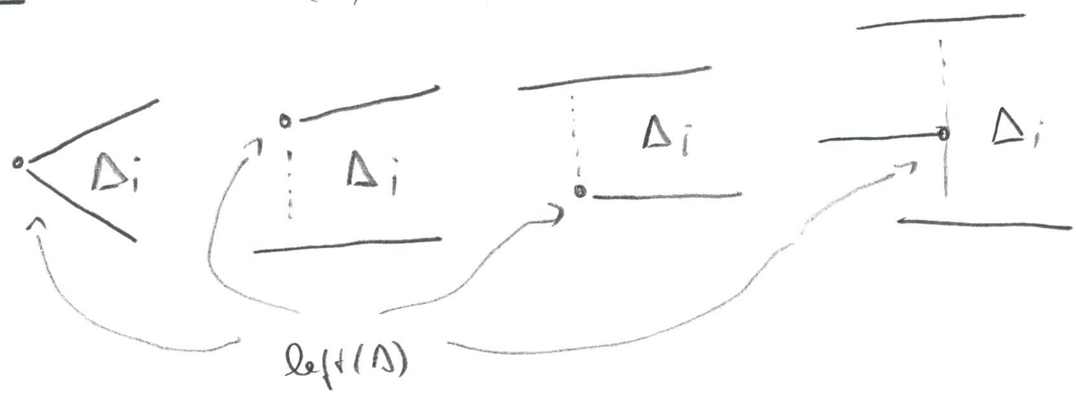
(points to neighbors)

+ any C_i is a trapezoid/triangle
+ $\underline{\underline{O(n)}}$ additional segments
+ C_i defined by left/right point and 2 segments
+ Two or one neighbor
+ Pointer to neighbors!

Trapezoidal map



4 cases for left(Δ)



Lemma 48: The trapezoidal map of a set of n segments in general position has $O(n)$ size.

Proof (Exercise, more precisely)

Randomized incremental algorithm

- Build bounding box of $S = \{s_1, \dots, s_n\}$
- Compute random permutation say $s_1, \dots, s_n$ of S
- Find trapezoid intersected by s_i: $\Delta_0, \Delta_1, \dots, \Delta_k$
- Remove trapezoids and replace them by new trapezoids

Incremental algorithm

$S_i = \{s_1, s_2, \dots, s_i\}$

$T(S_i)$ trapezoidal decomposition

$D(S_i)$ query structure for $T(S_i)$

directed acyclic graph DAG

Build $T(S)$, $D(S)$ by inserting $s_1, s_2, \dots, s_n$ in this order

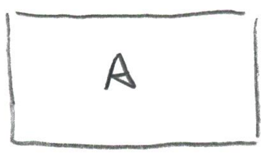
$T(S)$ always unique

$D(S)$ depends on insertion order

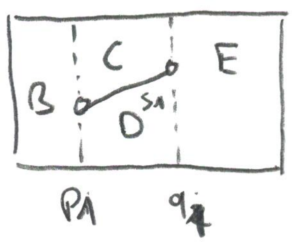
Example:

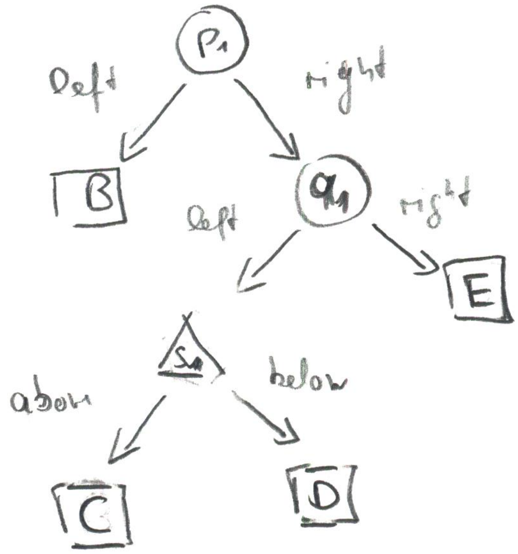
$T(S_i)$ $D(S_i)$

$i=0$

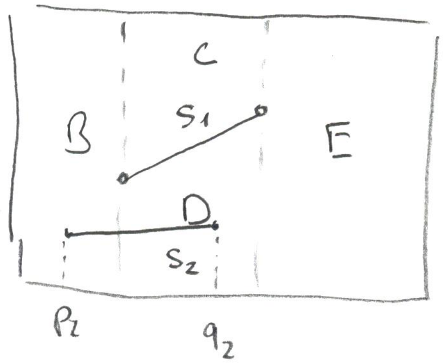


$i=1$





$i = 2$



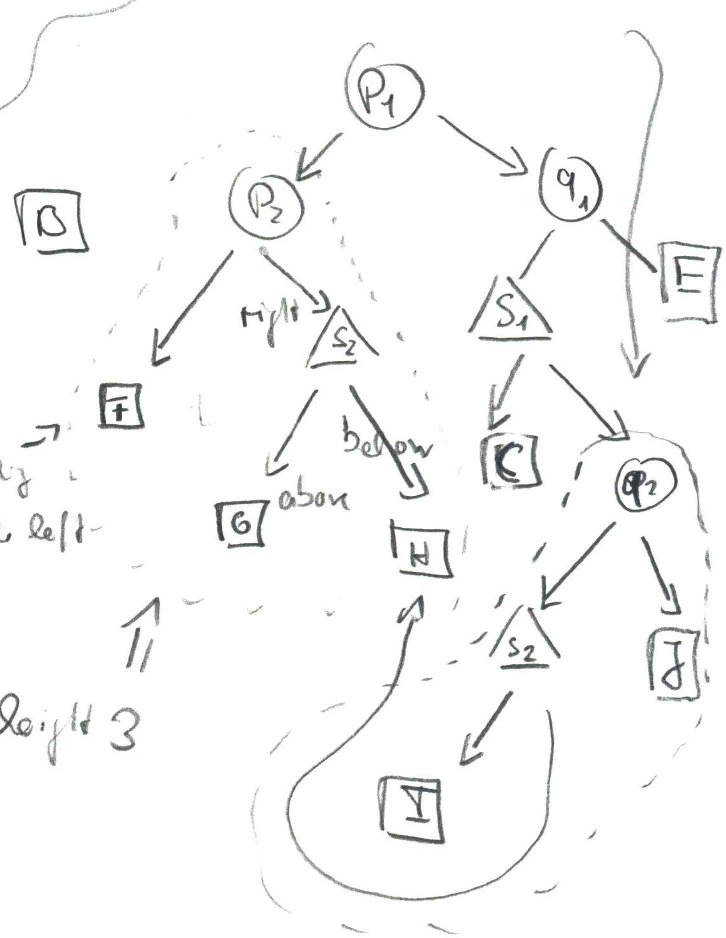
localize p_2 in $D(S_{i-1})$

- to the left of p_1 in B
- replace B
- trace s_2 through $D(S_{i-1})$ (neighboring trapezoids)
- insert q_2

neighbors of B

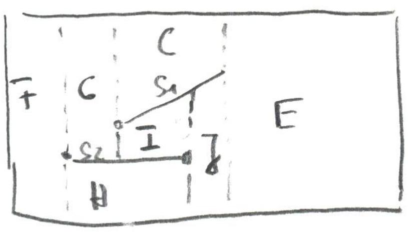
C, D below p_2

$\rightsquigarrow$ D ends there

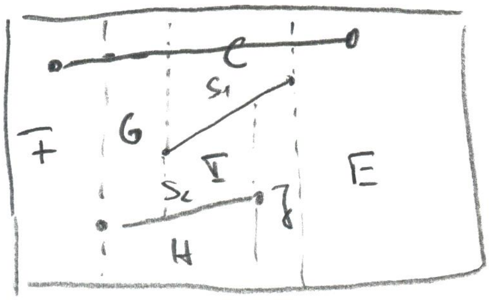


notes to the left new

height 3



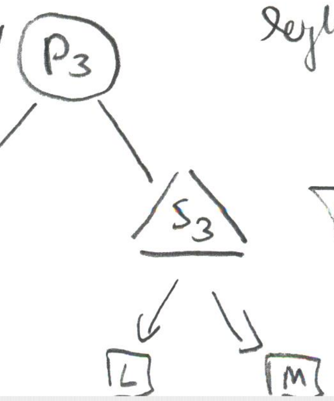
s_3 new



Replace:

F by

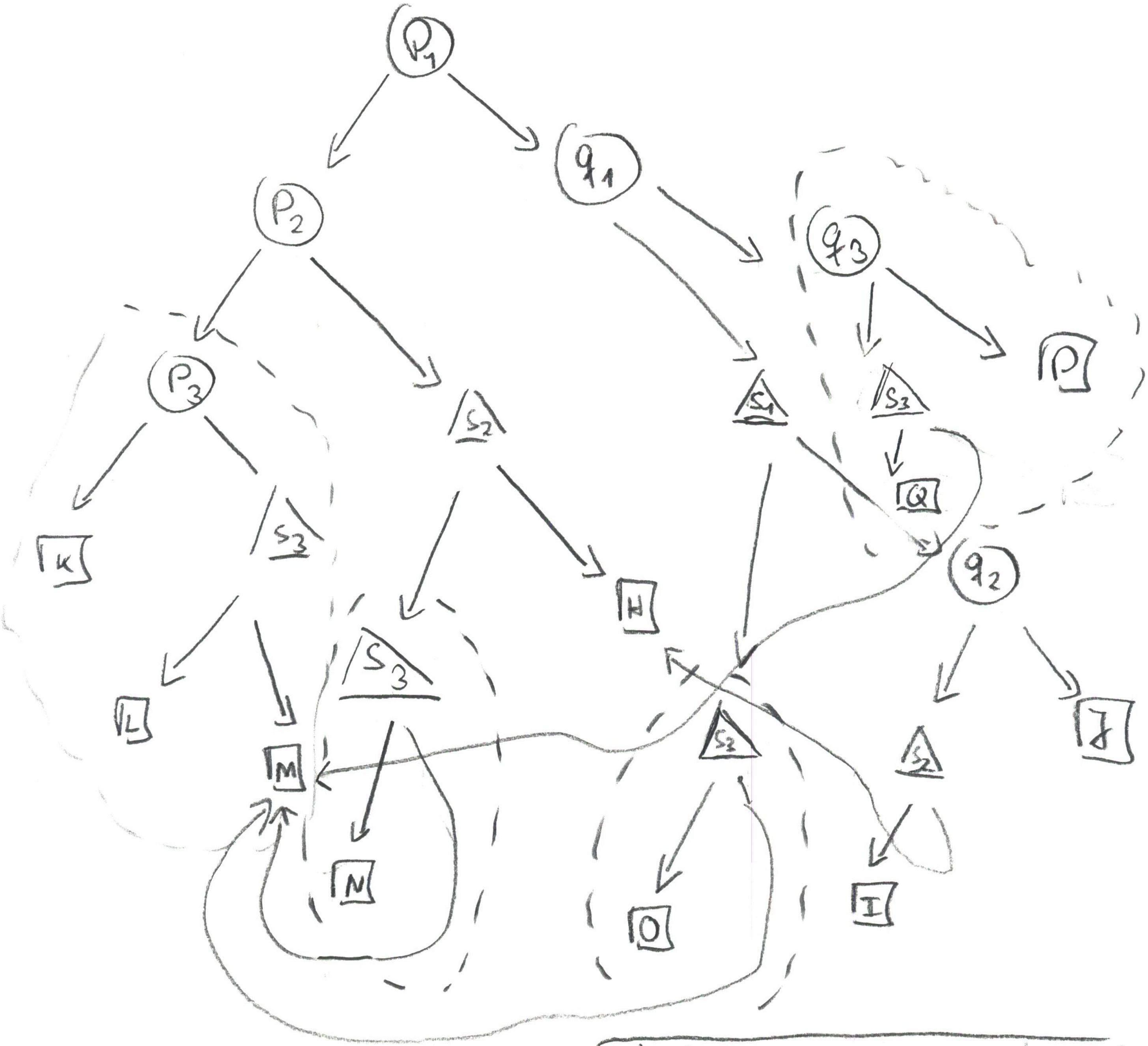
height 3

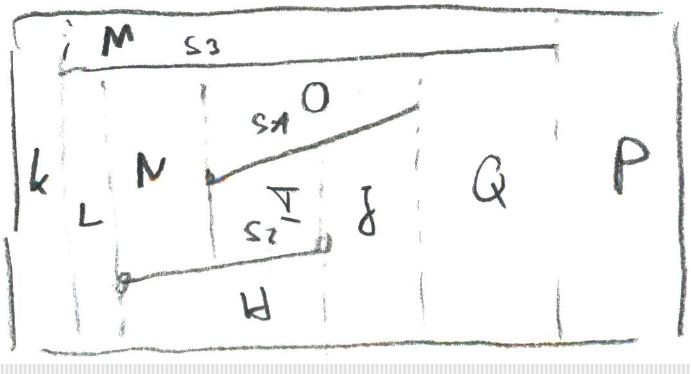


Trace s_3

Trace s_3: Splits Trace by neighbor pointers!

F, G, C, E





Facts:

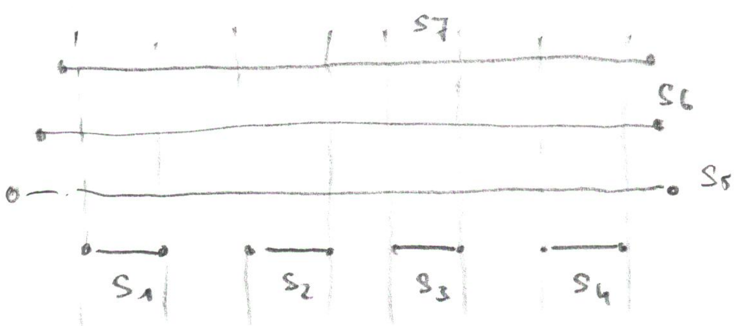
- $D(S_i)$ is a DAG
- some trapezoids of $T(S_i)$ will be destroyed | segment replaces trapezoid
- in $D(S_i)$ these trapezoids are replaced by small DAG's of height ≤ 3.

T independent from insertion order, D not!

Worst-Case performance

$D(S_n)$ can have quadratic size!

Example:



s_5, s_6, s_7

better than

s_7, s_6, s_5

always $\Omega(n)$ entries $\triangle_{S_i}$ replace $\boxed{T_i}$

$\Rightarrow \Omega(n^2)$! Size

Query-structure stores the "history"!

Add substructures of height 3 $\leadsto$ height $D(S_n) \in O(n)$

Query: $O(n)$ time

Performance random insertion

Theorem 49 The randomized insertion algorithm computes a trapezoidal map for a set of n line segments (in general position) and a search structure in $O(n \log n)$ expected time. The expected size of $D(S)$ is in $O(n)$ and the expected query time is in $O(\log n)$.

Proof: First: Query time

x fixed Query point

Running time: Length of the search path in $D(S_i)$ to trapezoid Δ with $x \in \Delta$

Expected path length:

average query time for x for $n!$ possible insertion orders

Search path gets longer from S_{i-1} to S_i (at most by 3 edges!)

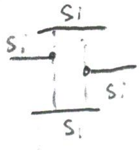
$\Rightarrow$ x-trapezoid in $T(S_{i-1})$ is not the same as in $T(S_i)$
(that contains x)

$\Rightarrow$ For x-trapezoid in $T(S_i)$ we have
bottom(x-trap.) $= s_i$ or top(x-trap.) $= s_i$
left(x-trap) $= p_i$ or right(x-trap.) $= q_i$

The probability that s_i intersects

x-trapezoid in $T(S_{i-1})$ in such a way is

$\frac{4}{i}$ (Backward analysis: Remove a segment s_i out of S_i. s_i was choosen uniformly at random. probability that s_i changes x-trapezoid in $T(S_{i-1})$ $\frac{4}{i}$)



$\Rightarrow$ expected path length (over all $n!$ possible insertion orders)

$$\leq \sum_{i=1}^{n} 3 \cdot \frac{4}{i} \leq 12 \ln n \quad O(\log n)$$

(Harmonic numbers)

$\uparrow$ (max new level)

(Backward argument: probability that s_i was choosen is $\frac{1}{i}$)

Next: <u>Space complexity</u>

Highest complexity in Structure $D(S_n)$

$\#$ leafs + $\underbrace{\#\text{ inner vertices}}_{\#\text{ of trapezoids in all earlier structures}}$ ("History")

$O(n)$ trapezoids

$k_i :=$ $\#$ of new trapezoids in $T(S_i)$ (may be $O(i)$ many)

$=$ $\#$ of trapezoids that vanishes after deleting s_i out of $T(S_i)$

Backwards analysis: S_i fixed, choose s_i uniformly at random for the delete operation!

Function for $\nabla(S_i)$, $s \in S_i$, $\Delta \in \nabla(S_i)$

$$\delta(\Delta, s) := \begin{cases} 1, & \Delta \text{ vanishes if } s \text{ is deleted} \\ 0, & \text{otherwise} \end{cases}$$

$$E(k_i) = \frac{1}{i} \underbrace{\sum_{\Delta \in \nabla(S_i)} \sum_{s \in S_i} \delta(\Delta, s)}_{\leq 4 \cdot |\nabla(S_i)| \in O(i)} \leq \frac{1}{i} \cdot C \cdot i \in O(1)$$

prob. that s_i is selected uniformly at random in S_i

(average over all)

Any trapezoid is bounded by at most 4 segments
Can only be deleted 4 times by some other!

$\Rightarrow$ expected size $O(n)$ $\quad \sum_{i=1}^{n} k_i \in O(n)$

Last: <u>build</u> complexity

Query for insertion of s_i $\quad O(\log i)$ exp. time

Insertion of s_i: $\quad E(k_i) \in O(1)$

$\Rightarrow O(n \log n)$ expected time

$\square$

<u>Next:</u> Worst Case!

Average query time! Worst Case query time of some points!

(Like: Degenerate Cases!)