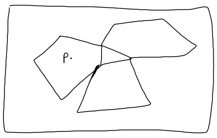
Point location

Given planar subdivision (map) give an algorithm that finds finds in which face a point lies.



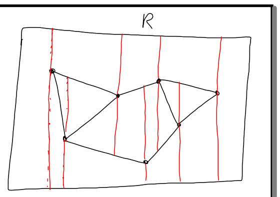
Idea: we construct a refinement of this map (not much larger than that of the original map) which is easier to search.
the faces will be trapezoids (with 2 vertical sides) or triangles (degenerate trapezoids) with one vertical side.

– This is called a trapezoidal map.

lis 15-15:47

Assumptions:

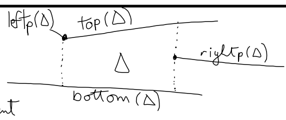
- set $S=\{s_1,\dots,s_n\}$ of n non-intersecting segments in the plane (except at endpoints)
- Enclosed in a box R
- No endpoints have same x-coordinate (remove this assumption later).



- From map S create trapezoidal map $T(S)$ by drawing a vertical line from each endpoint of a segment to the nearest upper segment & lower segment (possibly including the boundary of R)

lis 15-16:14

- top(Δ) segment of S bounding Δ from above.
- Likewise bottom(Δ) is segment of S bounding Δ from below.
- Left & right sides determined by endpoints of segments from S - leftp(Δ) & rightp(Δ).



- In case of triangle, left & right sides degenerate to a point.
- Trapezoid Δ is specified by (topΔ, bottomΔ, leftpΔ, rightpΔ)

lis 15-16:25

Theorem: Trapezoidal map for n segments has at most $6n+4$ vertices & $3n+1$ trapezoids.

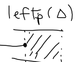
Proof:
- No of vertices: corners of R – 4
  - at most $2n$ endpoints, for each endpoint we have 2 new vertices

  So total $\leq 4+2n+2(2n) = 4+6n$.
- No. of trapezoids: count these by the no. with a given leftpoint.

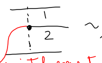
① 

~ 1

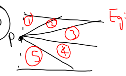
② leftp(Δ) is right endpoint of a segment



- each such right endpoint determines 1 face

③a 

$\sim$ determines 2 faces

③b 

Eg: 4 segments with common left endpoint p, determine 5 faces with leftpoint p.
k seg → $k+1$ faces.

k_i = no. of leftpoints which are common left endpoint of exactly i segments.
Then $n \geq k_1+2k_2+3k_3+\dots$

No. of trapezoids of type 3a & 3b
$2k_1+3k_2+4k_3\dots \leq 2(k_1+2k_2+3k_3\dots) \leq 2n$

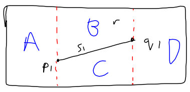
So grand total $\leq \underbrace{1}_{\text{bottom corner}} + \underbrace{n}_{\text{case 2}} + \underbrace{2n}_{\text{case 3}} = 3n+1$

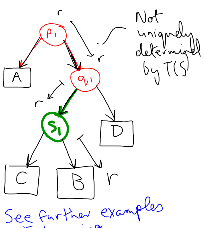
lis 15-16:33

Search structure

- Oriented graph $D(S)$ associated to $T(S)$:

directed graph (oriented)
- leaves are trapezoids (faces) of T(S).
- Inner nodes are endpoints of segments & segments: two edges from each inner node





– Given pt r we find face in which it lies by:
- If a node is an endpoint, go left if r lies to its left & right if r lies to its right. (Deg. case where we have equality will be dealt with later).
- If a node is a segment, go left if r lies below & right if r lies above.

See further examples in E-Learning.

lis 15-16:56

Randomised incremental algorithm

- Input $S=\{s_1,\dots,s_n\}$ set of segments.
- Randomise order.
- For $i=1,\dots,n$ we construct a trapezoidal map T_i & search structure D_i from T_{i-1} & D_{i-1} by adding a segment s_i.

3 steps
1. Find set $\Delta_0,\dots,\Delta_k$ of faces in T_{i-1} properly intersected by s_i.
2. Remove $\Delta_0,\dots,\Delta_k$ from T_{i-1} & replace them by new trapezoids appearing because of s_i.
3. Remove leaves $\Delta_0,\dots,\Delta_k$ from D_{i-1}, & replace those leaves by new subgraphs to create D_i.

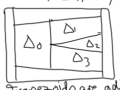
lis 15-17:11

① Following segment algorithm

See Fig 8.7

– Several cases: if leftpoint p of new segment s not an endpoint of $\{S_1, \ldots, S_{i-1}\}$ then it lies in interior of Δ_0.

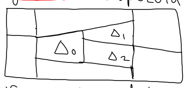
– Find Δ_0 by searching for p in $D(S_{i-1})$.

- To find $\Delta_1, \ldots, \Delta_K$ need notions of neighbour of trapezoid.
- 

Two trapezoids are adjacent if they share a vertical line, not a point.

– Δ_0, Δ_1 adj, Δ_0, Δ_3 are adj.,

Δ_0 & Δ_2 are not.



Δ_0 is lower left neighbour of Δ_2, & Δ_2 lower right neighbour of Δ_0.

– If 2 adj. trapezoids have common bottom say one is lower left neighbour, & one is the lower right neighbour.

– . . . . top say one is upper left neighbour / upper right neighbour.

lis 15-17:22

- Neighbours should be stored with trapezoidal map.
- Back to algorithm:

– if right endpoint q of s lies in Δ_0, then s lies completely in Δ_0

– Otherwise s intersects right upper / right lower neighbour of Δ_0 – if rightp(Δ_0) is above s it is lower right neighbour otherwise . . . . . upper right neighbour

– Continue in this way, find $\Delta_0, \Delta_1, \ldots, \Delta_K$. See Fig 8.7.

– Also consider case where p is endpoint of some $\{S_1, \ldots, S_{i-1}\}$. See E-Learning.

lis 15-17:37