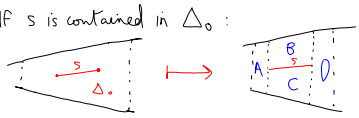
Steps 2&3 - brief outline (geometric)

- If s is contained in Δ_0:

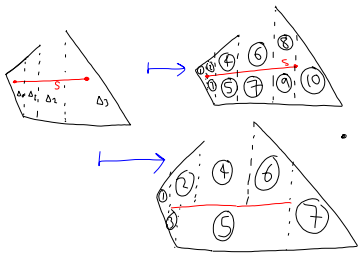


See E-Learning Fig 8.13 & 8.14 for update of search str.

Otherwise ↓

lis 22-16:17

- Otherwise, draw vertical lines from endpoints p, q of s to nearest segments (only if endpoints weren't present).
- New vert. lines, together with s, naturally split $\Delta_0, \ldots, \Delta_k$ into smaller trapezoids.
- Merge adjacent trapezoids with same top & bottom.
- See Fig 8.15 & 8.16 in E-Learning for update of the search str.



lis 22-16:21

Search time to find a point - $O(\log n)$

- Search structure created in n steps.
- Consider point r.
- X_i : no. of nodes added to search path for r at i'th stage.
- From 8.14, 8.16 we see $X_i \leq 3$.
- Expected search time:

$$\sum_{i=1}^{n} E(X_i) \quad \text{where } E(X_i) = \text{expected no. of nodes added at } i\text{'th stage to search path for } r.$$

- $E(X_i) = 0 \cdot p(X_i = 0) + 1 \cdot p(X_i = 1) + 2 \cdot p(X_i = 2) + 3 \cdot p(X_i = 3)$
  $\leq 3 \cdot p(X_i \neq 0)$
- Let $\Delta_i \in T_i$ be the trapezoid r in the i'th trapezoidal map T_i
- $p(X_i \neq 0) = p(\Delta_i \neq \Delta_{i-1})$
  $\leq p(\text{top}\,\Delta_i \neq \text{top}\,\Delta_{i-1}) + p(\text{bottom}\,\Delta_i \neq \text{bottom}\,\Delta_{i-1})$
  $+ p(\text{left}_p\,\Delta_i \neq \text{left}_p\,\Delta_{i-1}) + p(\text{right}_p\,\Delta_i \neq \text{right}_p\,\Delta_{i-1})$

lis 22-16:33

- $\text{top}\,\Delta_i \neq \text{top}\,\Delta_{i-1} \iff s_i = \text{top}\,\Delta_i$ which has probability $1/i$.
- Sim. for the bottom.
- $p(\text{left}_p\,\Delta_i \neq \text{left}_p\,\Delta_{i-1})$
  $= p(\text{left}_p\,\Delta_i$ is an endpoint of s_i & none of $s_1, \ldots, s_{i-1})$
- Since $p(\text{left}_p\,\Delta_i$ an endpoint of s_i & no others$)$
  $p(\text{left}_p\,\Delta_i$ ------ s_i & no others$) \leq 1$
- Sim for right we see $p(\text{left}_p\,\Delta_i \neq \text{left}_p\,\Delta_{i-1}) \leq 1/i$
- Hence $p(X_i \neq 0) \leq 4/i$.

lis 22-16:45

So $\sum_{i=1}^{n} E(X_i) \leq \sum_{i=1}^{n} 3\,p(X_i \neq 0)$

$\leq \sum_{i=1}^{n} 3\left(4 \cdot \frac{1}{i}\right) \leq 12\left(1 + \sum_{i=2}^{n} \frac{1}{i}\right)$

$\leq 12\left(1 + \int_1^n \frac{1}{x}\,dx\right)$

$= 12(1 + \log n) = O(\log n)$

lis 22-16:51

Expected size of search structure is $O(n)$

Proof: Size = no. of leaves + no. of internal nodes

$\leq (3n+1) + \sum_{i=1}^{n}$ (no. of internal nodes arising at i'th step)

$= (3n+1) + \sum_{i=1}^{n} (u_i - 1)$

where u_i = no. of new trapezoids arising at stage i.

So randomized complexity $\leq O(n) + \sum_{i=1}^{n} E(u_i)$.

To complete proof, must show $E(u_i) = O(1)$.

lis 22-16:55

- Must show $E(u_i) = O(1)$ where u_i = no. of trapezoids added at ith step.
- For $\Delta \in T(S_i)$ & a segment $s \in S_i$ let $\lambda(\Delta, s) = \begin{cases} 1 & \text{if } \Delta \text{ disappears from } T(S_i) \text{ when we remove } s \text{ from } S_i \\ 0 & \text{otherwise.} \end{cases}$
- Since Δ determined by at most 4 segments from S_i, so $\sum_{s \in S_i} \lambda(\Delta, s) \leq 4$.
- Now $u_i = \sum_{\Delta \in T(S_i)} \lambda(\Delta, s_i)$ (no of trapezoids at ith stage)
- Now $\sum_{\Delta \in T(S_i)} \sum_{s \in S_i} \lambda(\Delta, s) \leq \sum_{\Delta} 4 \leq 4 |T(S_i)| \leq 4(3i+1) = O(i)$.

$\sum_{s \in S_i} \sum_{\Delta \in T(S_i)} \lambda(\Delta, s) =$ no. of trapezoids disappearing when we remove s_i + no of trap. . . . . s_i

Since the $s_1, \ldots, s_n$ are in random order, $E(u_i) = \frac{1}{i} \left(\sum_{s} \sum_{\Delta} \lambda(\Delta, s) \right) \leq \frac{O(i)}{i} = O(1)$.

lis 22-17:01

Theorem Alg. constructs search structure in expected time $O(n \log n)$

Proof Time for creating $T(S_i)$ & $D(S_i)$ from $T(S_{i-1})$ & $D(S_{i-1})$ is

$E(u_i)$ no of trapezoids added + time searching in $D(S_{i-1})$ for trapezoid containing left endpoint of s_i

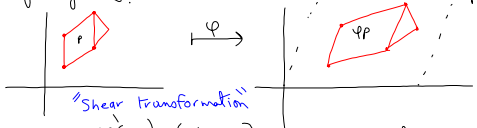
So time $\leq \sum_{i=1}^{n} (E(u_i) + O(\log i)) \leq n\, O(1) + n O(\log n)$

$= O(n \log n)$

lis 22-17:15

Removing restrictive assumptions

- No endpoints of segments have same x-coord.
- Search point p has different x-coord to all endpoints of segments.



$\varphi(x,y) = (x + \varepsilon y, y)$ for very small ε

- Doing this, no 2 segments in φS have same x-coord as endpoints. Likewise φp has different x-coords to endpoints of φS.
- Searching for φp in φS will produce the desired face – so run same algorithm on φS & φp.

lis 22-17:23

- In fact, do not need to do shear transformation at all.
- For φp lies to left of $\varphi q \iff p_x < q_x$ or $p_x = q_x$ & $p_y < q_y$.
  $\iff$ $p < q$ in lexicographic order.
- So it suffices to modify algorithm using lex order instead of checking if points lie to left or right.
- This gives same result but removes restrictive assumptions.

lis 22-17:37