<u>Dnes</u>. ~~Co se děje~~, když do $T\{s_1, \dots, s_{i-1}\}$ přidáme úsečku s_i.

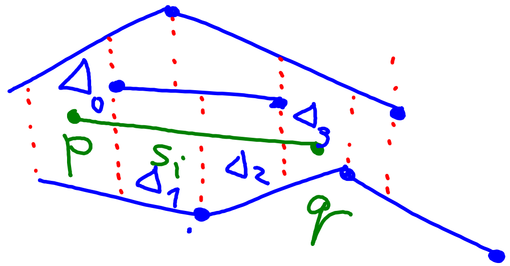
Krajní body úsečky s_i označíme p (levý) a q pravý.

Minule: jak najdeme lichoběžník Δ_0 ve kterém leží p.

(a) p není koncový bod $s_1, s_2, \dots, s_{i-1}$. Pak Δ_0 najdeme pomocí vyhledávací struktury $\mathcal{D}$.

(b)

Algoritmus pro nalezení lichoběžníků $\Delta_0, \Delta_1, \dots, \Delta_k$, které úsečka s_i protíná (FOLLOW SEGMENT)



Dnes: Co se děje, když do $T\{s_1, \dots, s_{i-1}\}$ přidáme úsečku s_i.

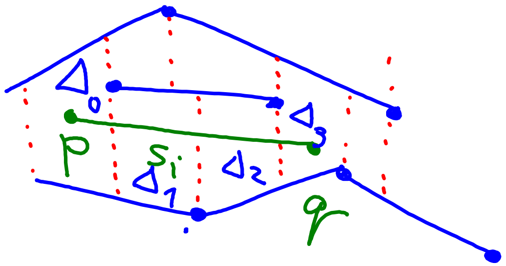
Krajní body úsečky s_i označíme p (levý) a q pravý.

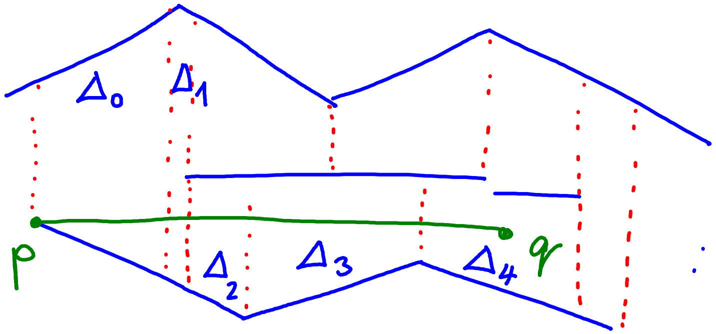
Minule: jak najdeme lichoběžník Δ_0 ve kterém leží p.

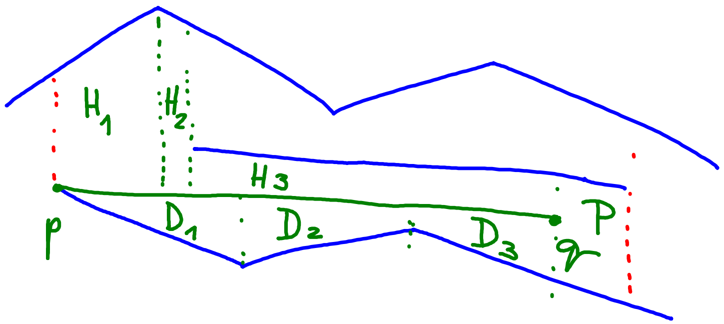
(a) p není koncový bod $s_1, s_2, \dots, s_{i-1}$. Pak Δ_0 najdeme pomocí vyhledávací struktury $\mathcal{D}$.

(b)

Algoritmus pro nalezení lichoběžníků $\Delta_0, \Delta_1, \dots, \Delta_k$, které úsečka s_i protíná (FOLLOW SEGMENT)







$\Delta_0, \Delta_1, \Delta_2, \Delta_3, \Delta_4$ vyndáme z $T(S_{i-1})$

a místo nich vložíme

$H_1, D_1 \quad D_2, H_3$

Tím dostaneme $T(S_i)$.

lokalizace_bodu.pdf

slajdy 19,21 - přesný popis algoritmu

<u>Věta</u> Očekávaná velikost vyhledávací struktury ~~je~~ $\mathcal{D}$ je $O(n)$, očekávaný čas konstrukce je $O(n \log n)$ a očekávaný čas vyhledávání je $O(\log n)$.

Jak odstranit předpoklad, že žádné dva body (koncové body úseček + bod q, jehož polohu hledáme) nemají stejné souřadnice x.

Pomocí tzv. shear transformation

$$\varphi: \begin{pmatrix} x \\ y \end{pmatrix} \longmapsto \begin{pmatrix} x + \varepsilon y \\ y \end{pmatrix} \quad \text{kde } \varepsilon > 0 \text{ malé}$$

jestliže $p_x = q_x$ a $p_y < q_y$, pak $\varphi(p)_x < \varphi(q)_x$.

$$\varphi(p)_x = p_x + \varepsilon p_y < q_x + \varepsilon q_y = (\varphi(q))_x$$

Platí: $\varphi(p)_x < \varphi(q)_x$ právě když $p_x < q_x$, nebo
$p_x = q_x$ a $p_y < q_y$

tj. právě když $p < q$ v lexikografickém uspořádání

Není potřeba $\varepsilon > 0$ pro transformaci hledat, stačí používat při porovnání „podle souřadnice x" lexikografické uspořádání. Podle souřadnice y používáme standardní uspořádání

$$X_i \neq 0 \iff \Delta^q(\mathcal{S}_{i-1}) \neq \Delta^q(\mathcal{S}_i)$$

$$E(X_i) = 0 \cdot p(X_i = 0) + 1 \cdot p(X_i = 1) + 2 \cdot p(Y_i = 2) + 3 \cdot p(X_i = 3)$$

$$\leq 3 \cdot p(X_i \neq 0) = 3 \cdot p(\Delta^q(\mathcal{S}_i) \text{ změní při odstranění } s_i)$$

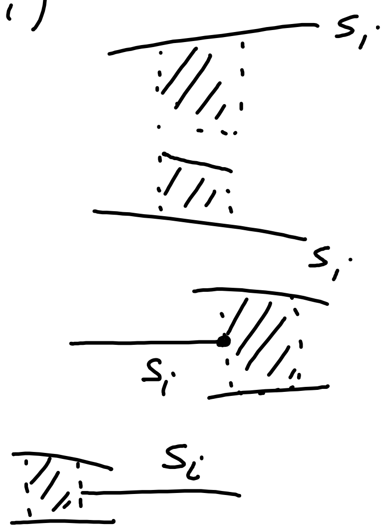
$\Delta^q(\mathcal{S}_i)$ změní při odstranění s_i z $T(\mathcal{S}_i)$

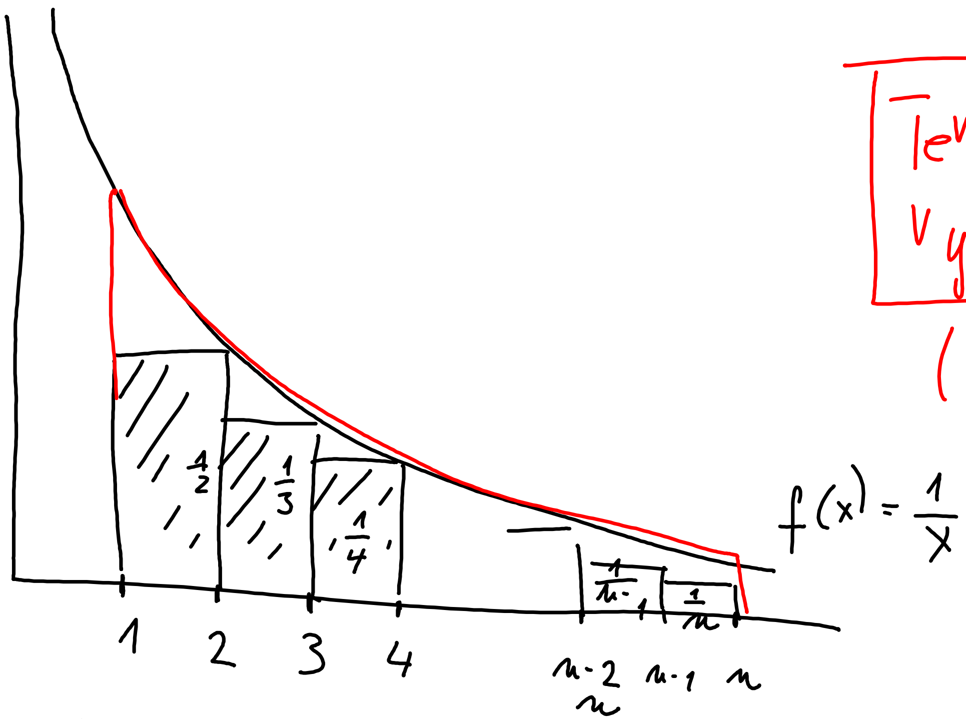
jestliže (1) s_i je vršek Δ^q

nebo (2) s_i je spodek Δ^q

nebo (3) s_i vyplňuje levý bod Δ^q

nebo (4) s_i vyplňuje pravý bod Δ^q





Tento důkaz budu vyžadovat u zkoušky!

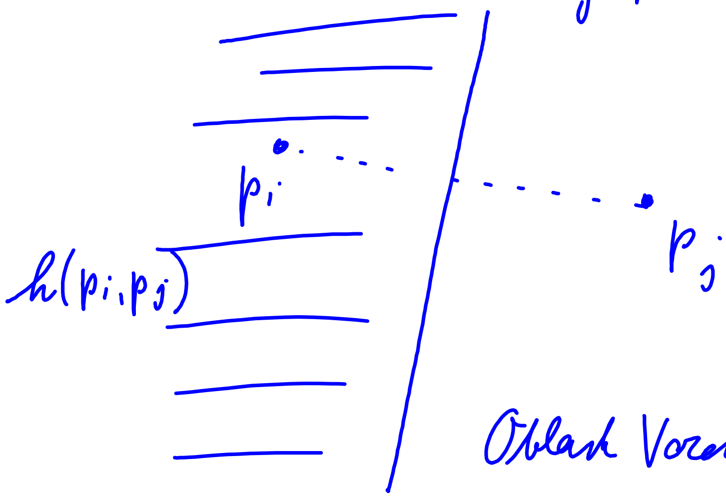
(Myslím poslední 4 řádky)

$$\frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \leq \int_1^n \frac{1}{x}\,dx = \left[\log x\right]_1^n = \log n$$

Pro body p_i a p_j je množina bodů q takových, že

$$\mathrm{dist}(q, p_i) = \mathrm{dist}(q, p_j)$$

přímka = osa úsečky p_i, p_j



$$h(p_i,p_j) = \{ q \in \mathbb{R}^2, \ \mathrm{dist}(p_i,q) \le \mathrm{dist}(p_j,q) \}$$

polorovina

Máme množinu

$$P = \{p_1, p_2, \dots, p_n\}$$

Oblast Voronoia bodu p_i je

$$V(p_i) = \bigcap_{j \neq i} h(p_i,p_j)$$

Věta. Počet vrcholů diagramu Voronoi-a pro n bodů je $\leq 2n - 5$ a počet hran je $\leq 3n - 6$.

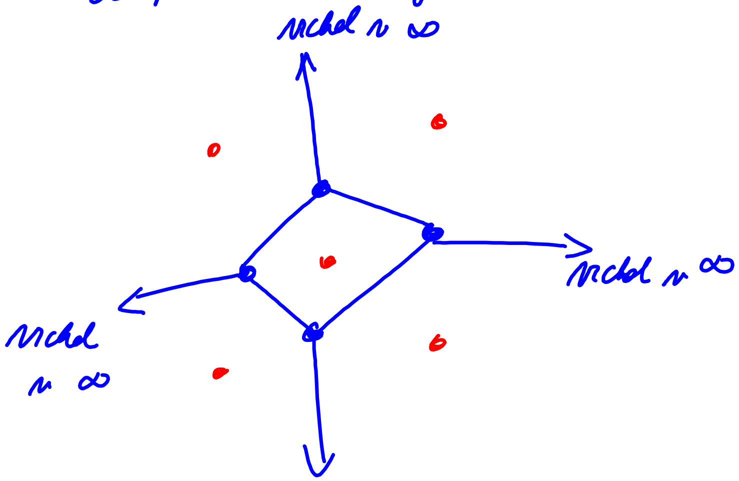


Diagram Voronoia je neorientovaný rovinný graf (souvislý). Proto pro něj platí Eulerova věta

$$(n_v + 1) - n_e + n_f = 2$$

počet vrcholů, hran, oblastí

virtuální bod v ∞

$n_f = n$