TRIANGULACE MNOHOÚHELNÍKŮ

má 2 kroky

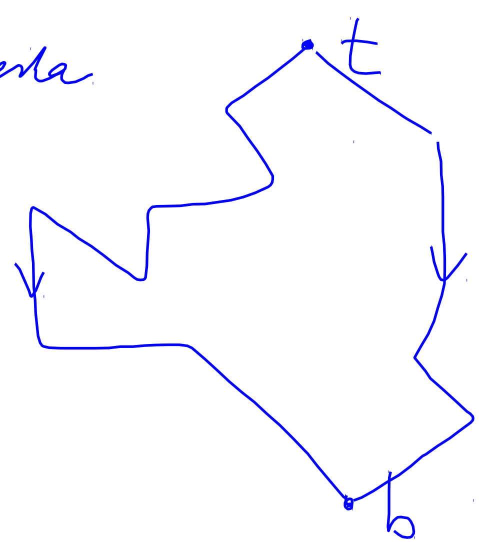
(1) rozdělení na monotónní mnohoúhelníky

(2) triangulace mon. mnohoúhelníka

Lexikografické uspořádání

$$p > q \iff p_y > q_y \text{ nebo } p_y = q_y \text{ a } p_x < q_x$$

levá cesta



pravá cesta

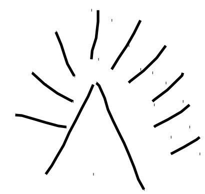
mnohoúhelník je monotónní

$\iff$ obě cesty klesající

Pomocí zametací přímky

Události – vrcholy mnohoúhelníka, několik typů

Důvod, proč mnohoúhelník není monotónní, je výskyt vrcholů typu split

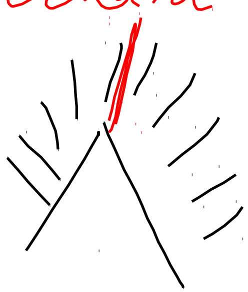


a typu merge

Korektnost algoritmu

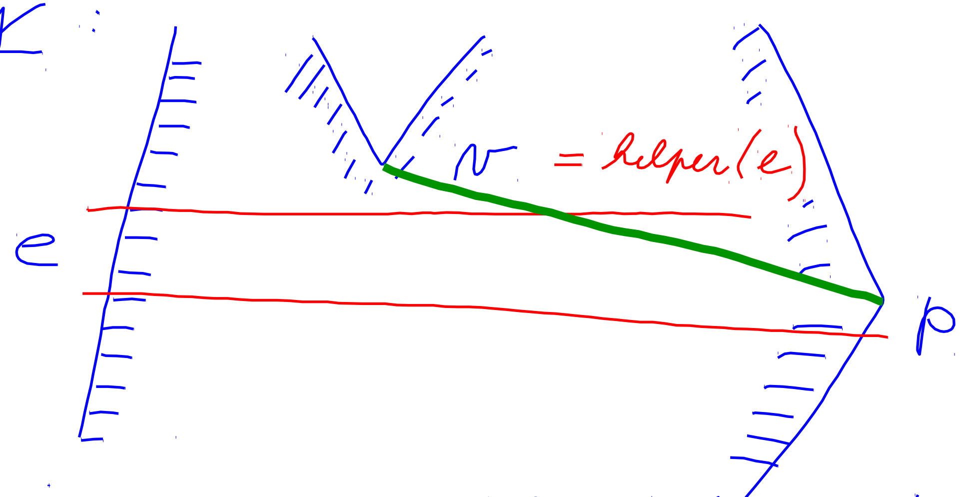
– algoritmus odstraní všechny vrcholy typu split a merge.

Split vrcholy – jednoduché, neboť se odstraní v okamžiku, kdy jimi prochází zametací přímka



3

Merge vrcholy:



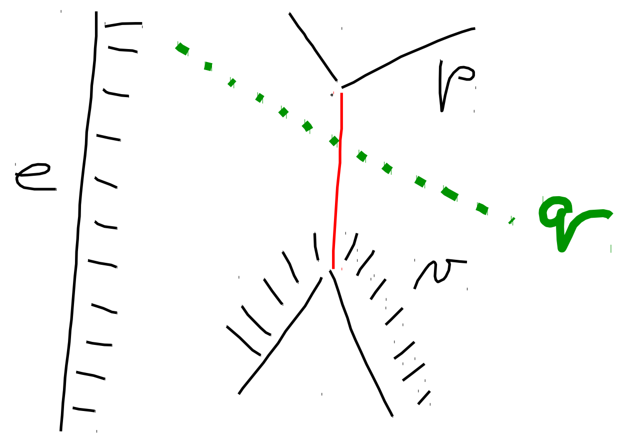
Vrchol v jako merge vrchol odstraníme v okamžiku, kdy zametací přímka prochází vrcholem p, když v roli pomocníka hrany e vystřídá vrchol v

– Jistě je potřeba se přesvědčit, že se úhlopříčky přidané algoritmem neprotínají.

Indukcí: Před tím než se zametací přímka dostane do vrcholu v, se žádné přidané úhlopříčky neprotínají. Ukážeme, že nové úhlopříčky vedoucí z v směrem nahoru žádné další

Také nepěkné. To bychom měli dělat pro každý typ vrcholu zvlášť.

v split



Kdyby přidaná červená úhlopříčka protínala nějakou jinou, tak jeden z vrcholů těla jiné úhlopříčky by ležel pod p a byl nahrazen hranou e

Triangulace monotónního mnohoúhelníku

Zde máme klesající levou i pravou cestu.

Z těchto dvou uspořádání uděláme jediné.

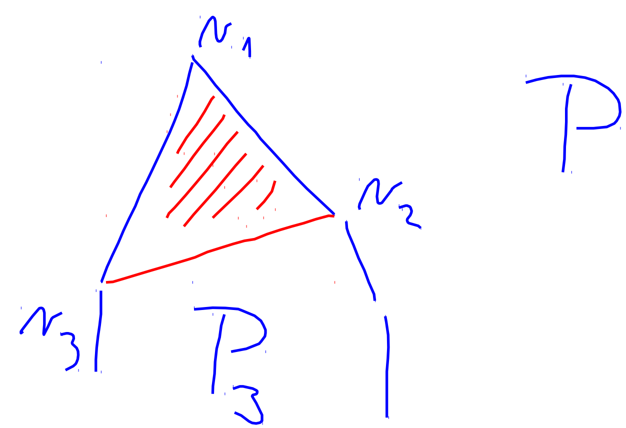
Opět metodou zametací přímky. Postupně od mnohoúhelníka „odřezáváme" trojúhelníky.

Pomocná struktura je zásobník, do kterého vrcholy ukládáme a zase vyndaváme.

Vrcholy označíme podle uspořádání $v_1 > v_2 > \dots > v_n$.

Do zásobníku vložíme (v_2, v_1)

Pokračujeme vrcholem v_3.



(1) v_3 leží na jiné cestě než (v_2, v_1)

v_3 spojíme s v_2, odstraníme $\triangle v_1 v_2 v_3$

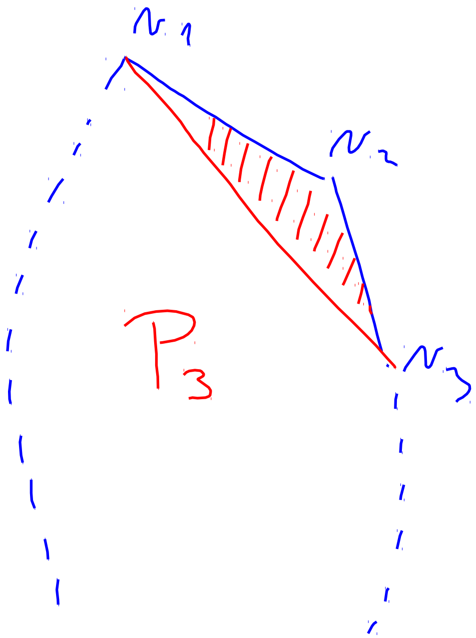
Z P zbyde P_3 - je opět monotónní

Zásobník $(v_2, v_1) \longrightarrow (v_3, v_2)$

(6)

(2) v_3 leží na stejné cestě jako $v_1 v_2$

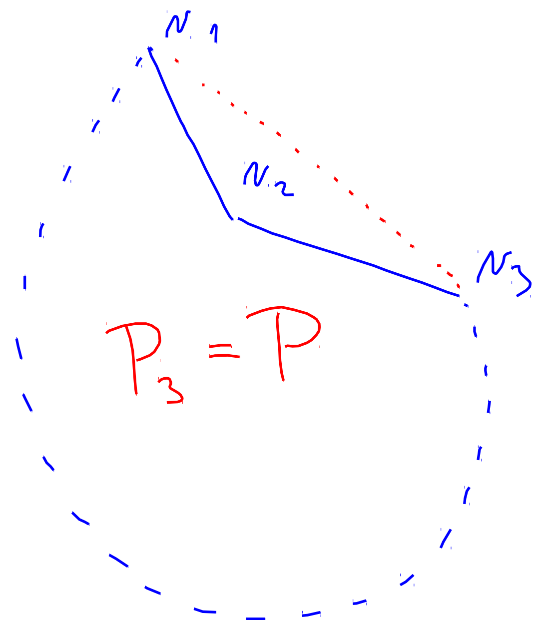
(a)



v_3 spojíme s

$$(v_2 v_1) \to (v_3 v_1)$$

(b)



$$(v_2, v_1) \mapsto (v_3 v_2 v_1)$$

Přidání vrcholu v_j, $j \geq 4$

Zásobník $\left(v_{j-1} \; v_{i_{k-1}} \quad \ldots \quad v_{i_2} \; v_{i_1} \right)$

$v_{j-1} = v_{i_k}$

$1 \leq i_1 < i_2 < \ldots < i_k = j-1$

Vrcholy vrcholy zásobníku leží na levé (nebo pravé) cestě mon. mnohoúhelníku P_{j-1}, který vznikl z P odstraněním trojúhelníků v předchozích krocích.

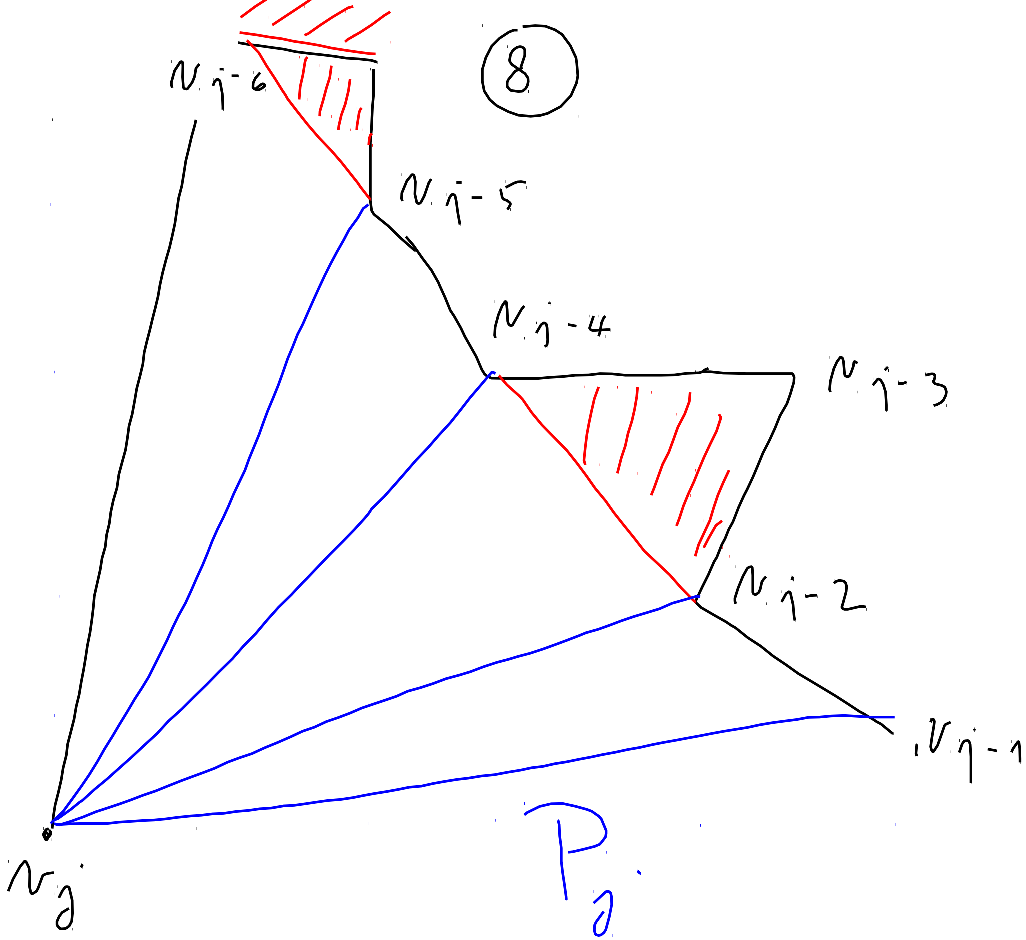
1. v_j je na druhé straně mnohoúhelníka P_{j-1}

v_j spojíme se všemi vrcholy v zásobníku

Odříznutím trojúhelníků získáme P_j

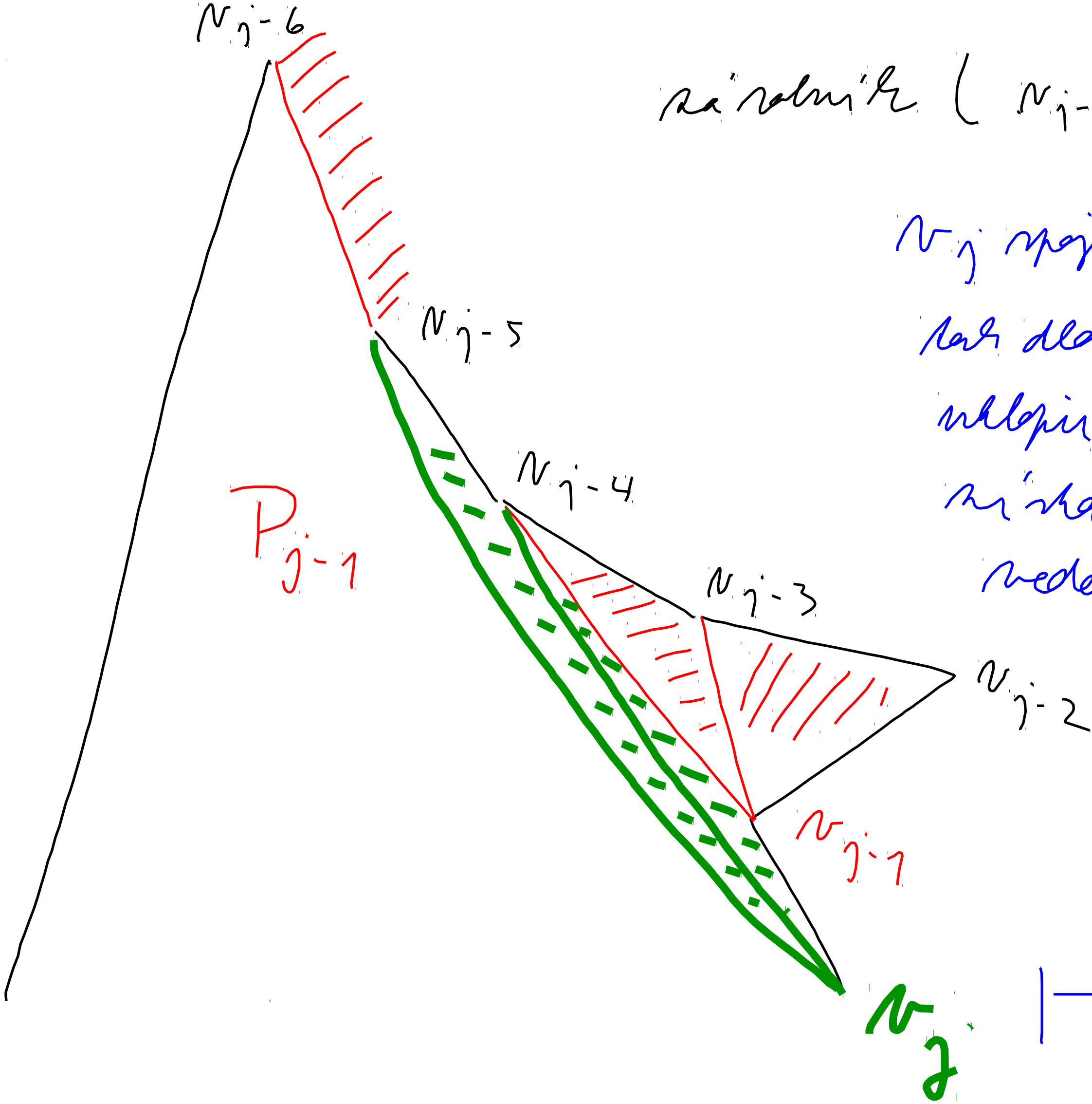
Zásobník $\longrightarrow (v_j \; v_{j-1})$

8



2 v_j leží na stejné straně jako vrcholy zásobníku

zásobník (v_{j-1}, v_{j-4}, v_{j-5}, v_{j-6})



v_j spojujeme s každým zásobníku, tak dlouho pokud získáváme úhlopříčky. Vzniklé Δ odřízneme, získáme P_j a vrcholy do kterých vedeme úhlopříčky odstraňujeme ze zásobníku. Pak do zásobníku dáme poslední spojující vrchol a v_j

$\longmapsto (v_j, v_{j-5}, v_{j-6})$

Průnik polorovin

V rovině máme n polorovin

$$H = \{h_1, h_2, \dots h_n\}$$

Chceme najít a sepsat $\displaystyle\bigcap_{i=1}^{n} h_i = \bigcap_{h_i \in H} h_i$

Musíme vyřešit, jak průniky polorovin reprezentovat.

~~Různé možnosti, jak průnik vypadá~~

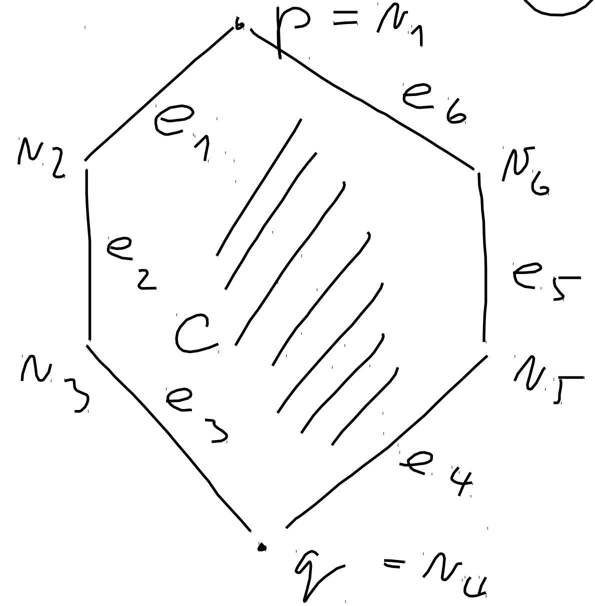
Lexikografické uspořádání

$$p > q \iff p_y > q_y \text{ nebo } p_y = q_y \text{ a } p_x < q_x$$

1. Průnik má maximální i minimální bod v tomto uspořádání.

Maximální p

Minimální q

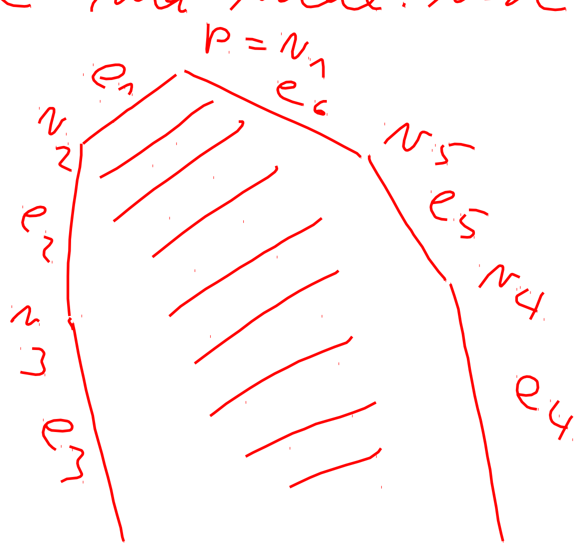


Hranice se dělí na dvě části

levá hranice $L(C) = (p = v_1, e_1, v_2, e_2, v_3, e_3, q = v_4)$

pravá hranice $P(C) = (p = v_1, e_6, v_6, e_5, v_5, e_4, q = v_4)$

2 Příklad C má max. bod, ale není minimální

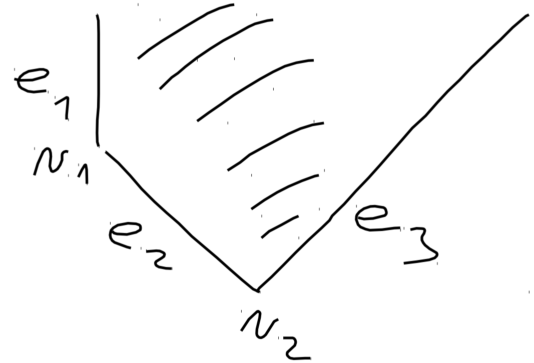


$L(C) = (v_1, e_1, v_2, e_2, v_3, e_3)$

$P(C) = (v_1, e_6, v_5, e_5, v_4, e_4)$

3 C obsahuje minimální bod, ale neobsahuje maximální bod

– analogicky



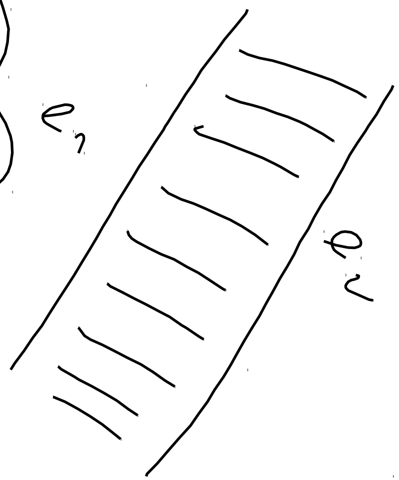
$L(C) = (e_1, v_1, e_2, v_2)$

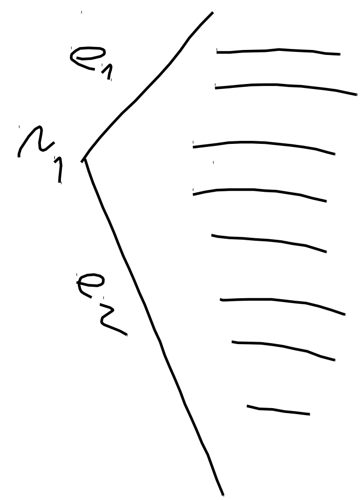
$P(C) = (e_3, v_2)$

4 C neobsahuje min. ani maximální bod

$L(C) = (e_1)$

$P(C) = (e_2)$





pravá nebo levá hranice je prázdná

$L(C) = (e_1, v_1, e_2)$

$P(C) = (\ )$

<u>Algoritmus</u> – rozděl a panuj

$$H = \{h_1, \dots h_n\}$$

vyjma zde dvě opačné

rozdělíme na 2 podmnožiny

$$H_1 = \{h_1, \dots h_{[\frac{n}{2}]}\} \qquad H_2 = \{h_{[\frac{n}{2}]+1}, \dots h_n\}$$

Spočtěme

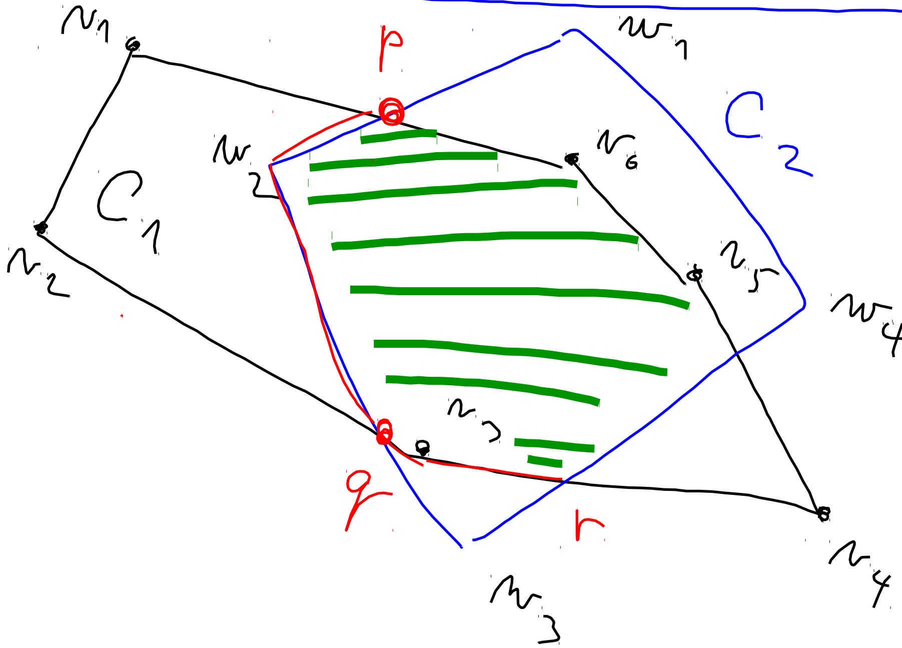
$$C_1 = \bigcap_{H_1} h_i \qquad C_2 = \bigcap_{h_i \in H_2} h_i$$

a chceme spočítat

$$C = C_1 \cap C_2$$

C_1 a C_2 máme popsány pomocí levých a pravých hranic. Máme $L(C_1)$, $P(C_1)$, $L(C_2)$ a $P(C_2)$ a chceme najít levou a pravou hranici průniku $L(C)$ a $P(C)$.

Jak vypadají vrcholy v $L(C_1 \cap C_2)$



- $w_2 \in L(C_2)$ a leží uvnitř C_1
- $v_3 \in L(C_1)$ a leží uvnitř C_2
- p míří z levé hranice C_2 a pravé hranice C_1
- q je průsečík levých cest
- r je průsečík levé hranice C_1 a pravé hranice C_2

Průnik $C_1 \cap C_2$ počítáme metodou zametací přímky.

Událostí jsou vrcholy C_1 a C_2 a k nim postupně přidáváme průsečíky hranic.

Na začátku seřadíme dle polohy vrcholy C_1 a C_2.

Binární vyvážený strom není potřeba, nahradíme ho pořadím hran (úsečka, polopřímka, přímka) zleva doprava, které protíná zametací přímka – a těch hran je nejvýše 4.

Při průchodu zametací přímky vrcholem řešíme tyto otázky. (Předpokládáme, že známe $L(C)$ a $P(C)$ nad vrcholem v)

- zda v patří do $L(C)$ nebo $P(C)$
- jak pokračuje $L(C)$ a $P(C)$ těsně pod v
- počítáme průsečíky hran procházejících vrcholem v se

se sousedními hranami. Pokud existují zařadíme je do fronty.