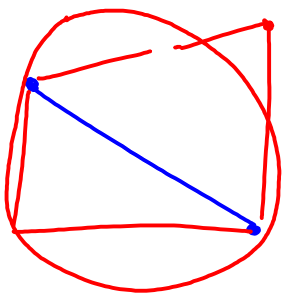
DELAUNAYOVA TRIANGULACE

Množina P bodů v rovině

- triangulace konv. obalu
- ideální cíl je dostat nějakou optimální triangulace
- prakticky dostaneme Delaunayovu triangulaci (= legální triangulaci) a v generickém případě i nějakou optimální triangulaci

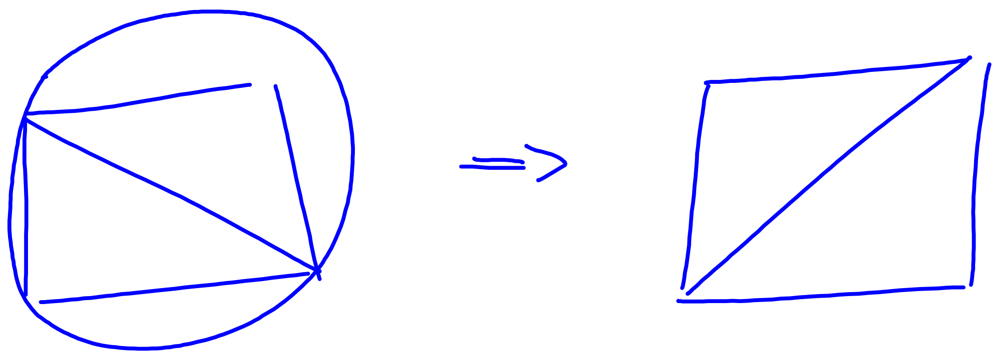
Legální triangulace se skládá pouze z legálních hran



legální hrana × ilegální

(2)

Flip



Po flipu je nová triangulace lexikograficky větší než původní
nebo
rovna

Triangulace je legální právě tehdy pro každou kružnice C opsanou △ $p_i p_j p_k$ neleží uvnitř bod p_l, kde $p_i p_j p_l$ je další △ triangulace.

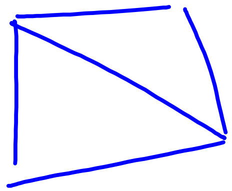
③

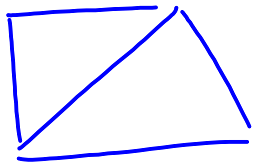
Delaunayova triangulace

Delaunayův graf = duální graf k diagramu Voronoia

$p_i p_j$ je hrana $\mathcal{D}g$ $\iff$ existuje kružnice C, p_i, p_j leží na C a žádný další bod z množiny P neleží na C ani uvnitř

pak-li v $\mathcal{D}g$ mnohoúhelníky, tak je jejich rozdělením na $\triangle$ vzniknou D. triangulace.

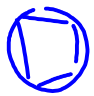




D. triangulace je určena jednoznačně

$\iff$ D. graf je triangulací

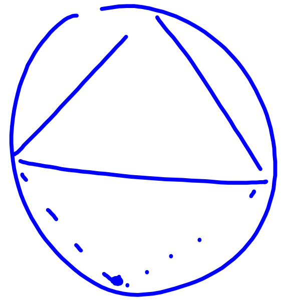
$\iff$ Žádné 4 body z P neleží na kružnici C, uvnitř které není žádný další bod z P



④

Hrana $p_i p_j$ $\mathcal{D}$ triangulace má vlastnost: Uvnitř kružnice opsané $\Delta p_i p_j p_k$ není další bod z množiny P.

Triangulace je Delaunayova: Uvnitř kružnice opsané $\Delta p_i p_j p_k$ neleží další bod z P.



Delaunayova triangulace

= legální triangulace

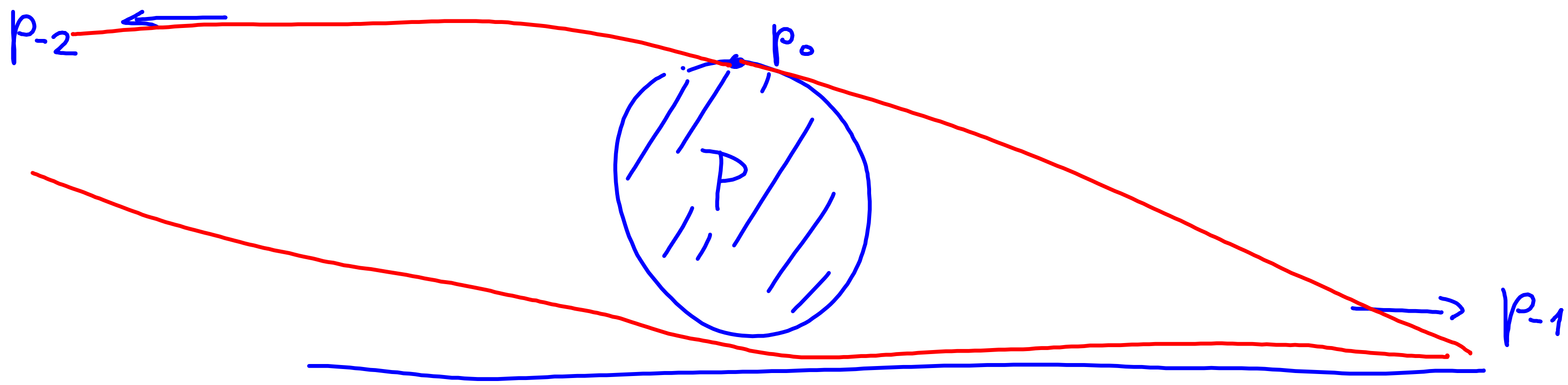
5

Algoritmus – přímočarý a nákladnější

$P = \{p_0, p_1, \ldots, p_n\}$

přičemž p_0 je maximální v lexikograf. uspořádání nejdříve podle y pak podle x.

Dodají se body p_{-1} a p_{-2} tak, že se vše odehrává v $\Delta p_0 p_{-1} p_{-2}$



6

Body p_{-1} a p_{-2} jsou voleny tak, že po nalezení D. triangulace $\Delta p_0, p_{-1}, p_{-2}$ s množinou $P \cup \{p_{-1}, p_{-2}\}$ a vypuštění bodu p_{-1} a p_{-2} a k nim přilehlých hran dostaneme D. triangulaci konv. obalu množiny $P = \{p_0, p_1, \ldots, p_n\}$

Algoritmus – náhodné uspořádání bodu $p_1, p_2, \ldots p_n$

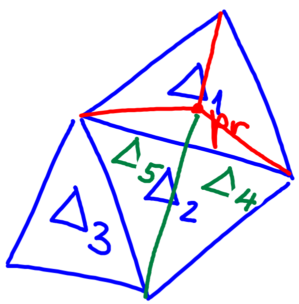
2 D. triangulace pro $\{p_{-2}, p_{-1}, p_0, p_1 \ldots p_{r-1}\}$ a vyhledávacího grafu pro tuto množinu, vytvoříme D. triangulaci pro $\{p_{-2}, p_{-1}, p_0, \ldots p_{r-1}, p_r\}$ a vyhledávací strukturu pro tuto triangulaci.

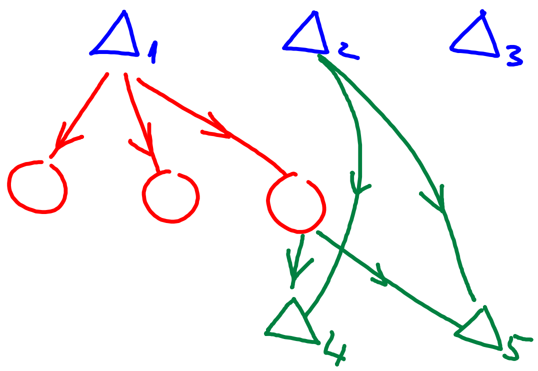
7

Co chceme po vyhledávací struktuře?

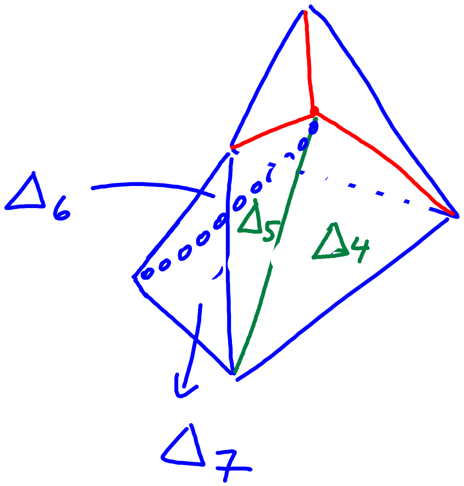
Umí zjistit, v kterém Δ dané D triangulace leží zadaný bod.

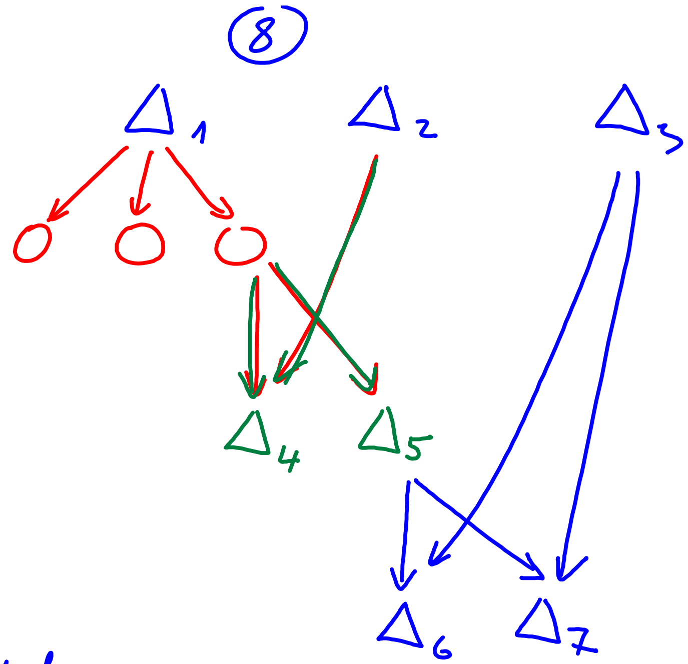
Je to orientovaný acyklický graf, listy jsou Δ aktuální D. triangulace, uzly jsou Δ předchozích triangulací.





listy grafu





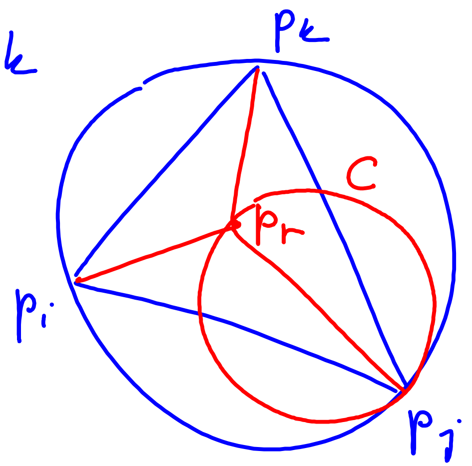
Tyto změny v triangulaci a ve vyhledávací struktuře je to, co provádí algoritmus pro přidání bodu p_r.

str. 36 a 37

9

Při legalizaci nové hrany vycházející z p_r jsou legální (dokonce Delaunayovy).

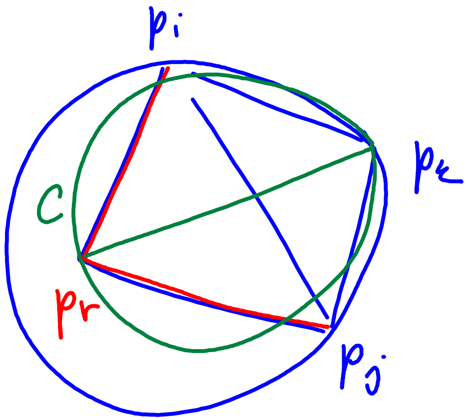
1. krok



Umíme C a na C neleží žádné další body z P.
$\Rightarrow p_r p_j$ je Delaunayova hrana = je legální

(10)

Další kroky



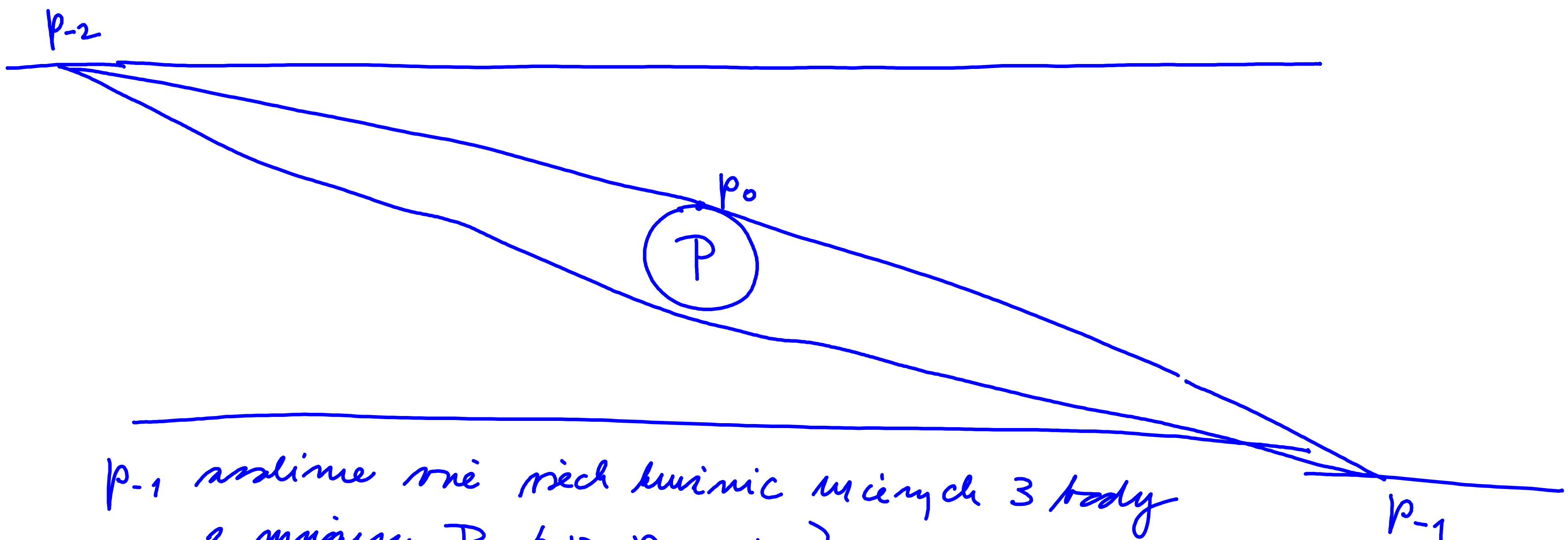
Nová hrana $p_r p_k$
kružnice C dotýkající
s kružnicí opsanou $\Delta p_k p_i p_j$
neobsahuje uvnitř ani na C
žádný další bod z P

$\Rightarrow p_r p_k$ Delaunayova hrana = legální

Věta
Očekávaný čas algoritmu
je $O(n \log n)$ a očekávaná velikost paměti je $O(n)$.

11

Volba bodů p_{-1} a p_{-2} a jak s nimi počítat

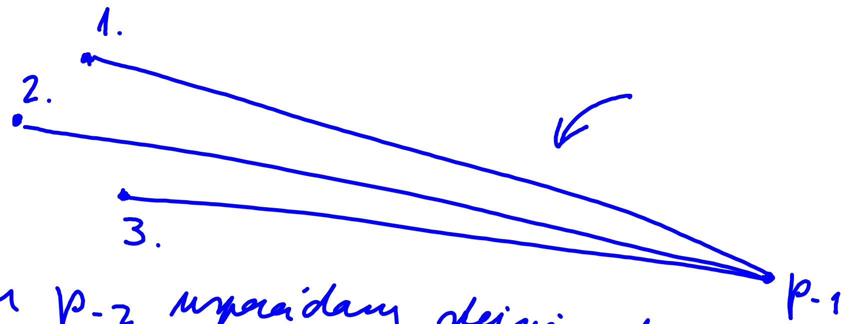


p_{-1} zvolíme vně všech kružnic určených 3 body z množiny $P = \{p_0, p_1, \dots p_n\}$, y-ová souřadnice < min y-ových souřadnic bodů z P

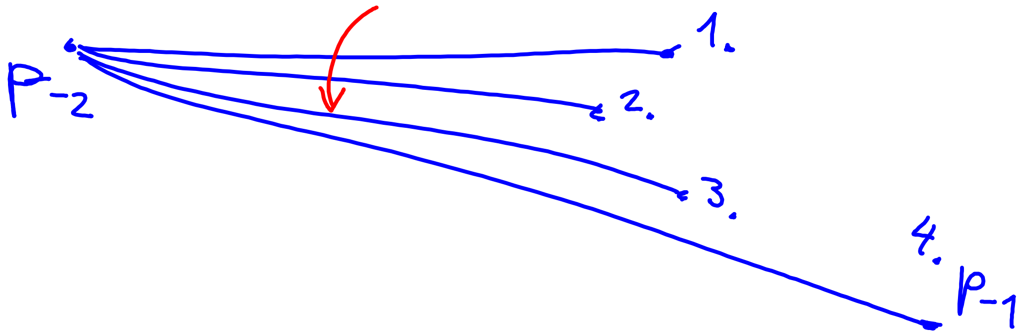
p_{-2} zvolíme nad P tak, že leží vně všech kružnic určených 3 body z $P \cup \{p_{-1}\}$

12

Body z P jsou kolem p_{-1} uspořádány stejně jako v jejich lexikografickém uspořádání:



Body z $P \cup \{p_{-1}\}$ jsou kolem p_{-2} uspořádány stejně jako v jejich lexikografickém uspořádání

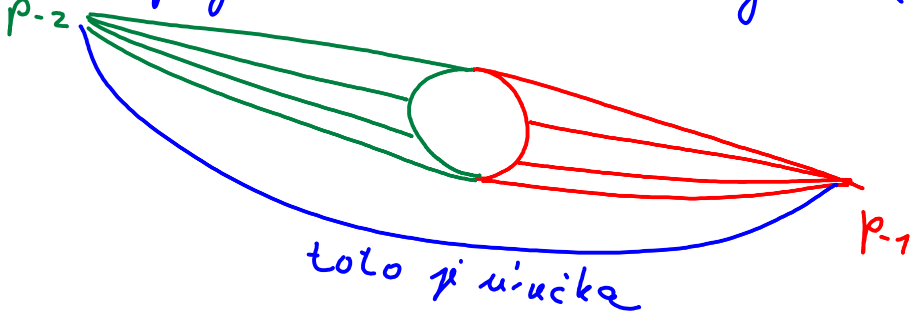


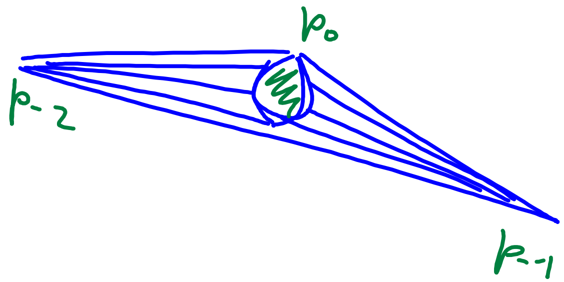
Speciálně platí, že P leží v $\triangle\, p_{-2}\, p_{-1}\, p_0$

13

Při této volbě se D triangulace $P \cup \{p_{-1}, p_{-2}\}$ skládá z

1. spojnic p_{-1} s body pravého konv. obalu
2. spojnic p_{-2} s body levého konv. obalu
3. z úsečky $p_{-2}p_{-1}$
4. z Delaunayovy triangulace množiny $P = \{p_0, \dots p_n\}$.

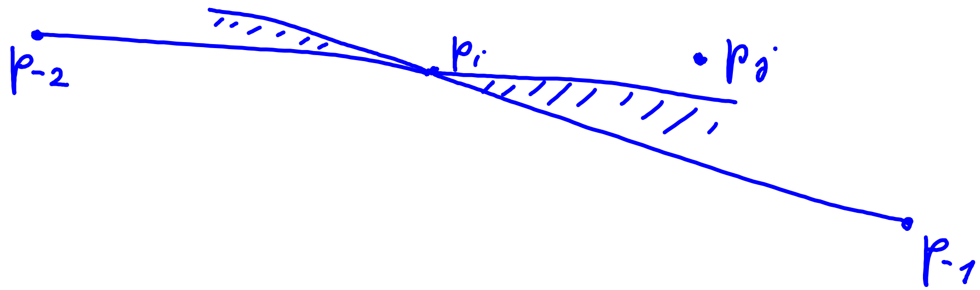




14

Při zjišťování, které hrany jsou ilegální, je potřeba zjišťovat polohu bodu p_j vzhledem k orientované přímce $p_i p_k$. K tomu pro $p_k = p_{-1}$ nebo p_{-2} použijeme následující ekvivalentní tvrzení:

- p_j leží vlevo od $\overrightarrow{p_i p_{-1}}$
- p_j leží vlevo od $\overrightarrow{p_{-2} p_i}$
- $p_j > p_i$ v lexikografickém uspořádání podle y a pak podle x



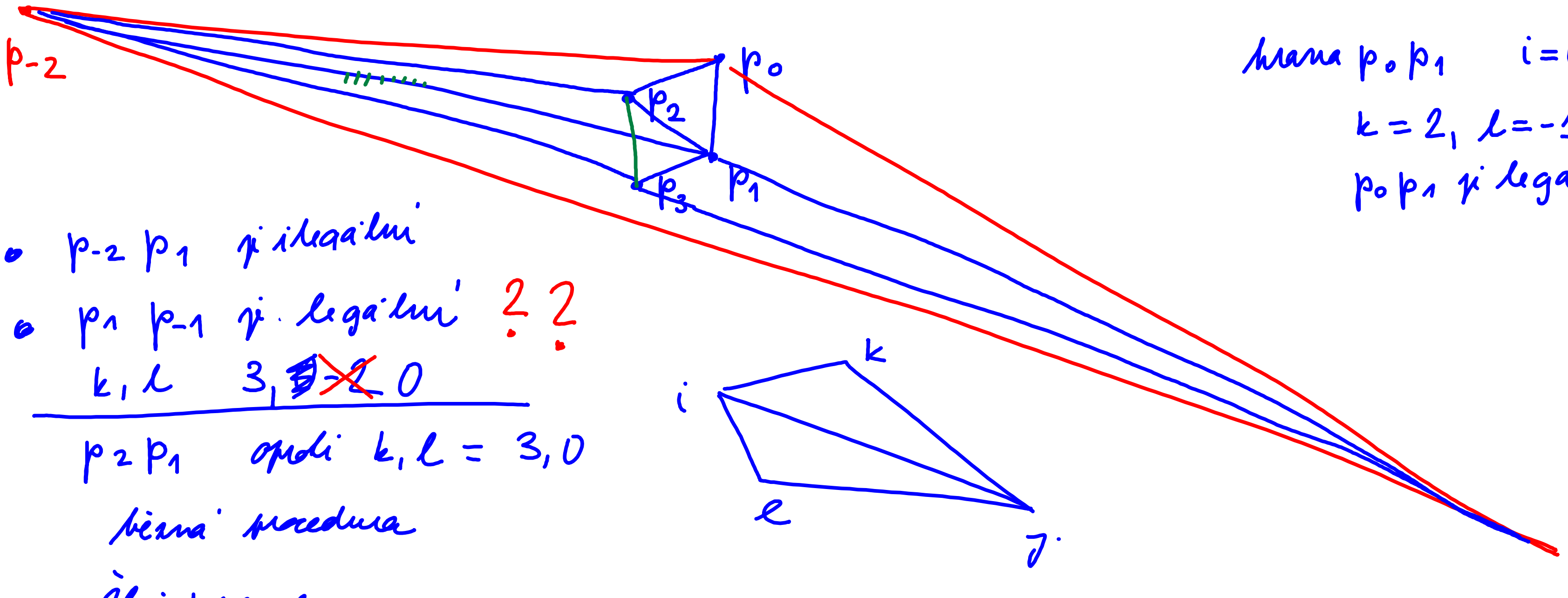
15

Ilegalita hrany obsahující bod p_{-1} nebo p_{-2}

- všechny hrany $\triangle\, p_0 p_{-1} p_{-2}$ jsou legální
- hrana $p_i p_j$ s přilehlými vrcholy p_k, p_l v případě, že nějaký index je < 0 a čtyřúhelník $p_k p_i p_l p_j$ je konvexní

  $p_i p_j$ je legální $\iff \min(k,l) < \min(i,j)$

Body p_{-1}, p_{-2} jsou fiktivní, jejich souřadnice neurčuji, počítáme s nimi podle předchozích pravidel.



brana $p_0 p_1$ $\quad i=0, j=1$

$k=2, \; l=-1$

$p_0 p_1$ je legalni

- $p_{-2} p_1$ je ilegalni
- $p_1 p_{-1}$ je legalni ? ?

$k, l \quad 3, \cancel{-2}\; 0$

$p_2 p_1$ obradi $k, l = 3, 0$

ista procedura

Četverokut $p_0 p_1 p_3 p_{-1}$ je nekonveksan
također p_1 i p_{-1} leže s iste strane od $\overrightarrow{p_3 p_0}$ $\Big\}$ $\Rightarrow$ $p_1 p_{-1}$ je legalni