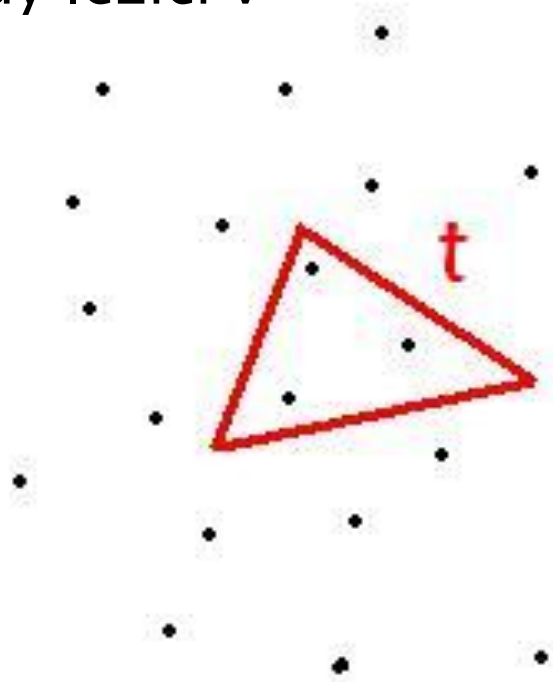
OI-OPPA. European Social Fund
Prague & EU: We invest in your future.

Cutting trees

Martin Chaloupka,
chaloma3@fel.cvut.cz,
ČVUT, fakulta elektrotechnická,
Praha

Cíl

- Vyřešit: Triangular range searching/counting problém
- Problém: Máme množinu S, obsahující n bodů na rovině a trojúhelník t, máme spočítat body ležící v trojúhelníku t

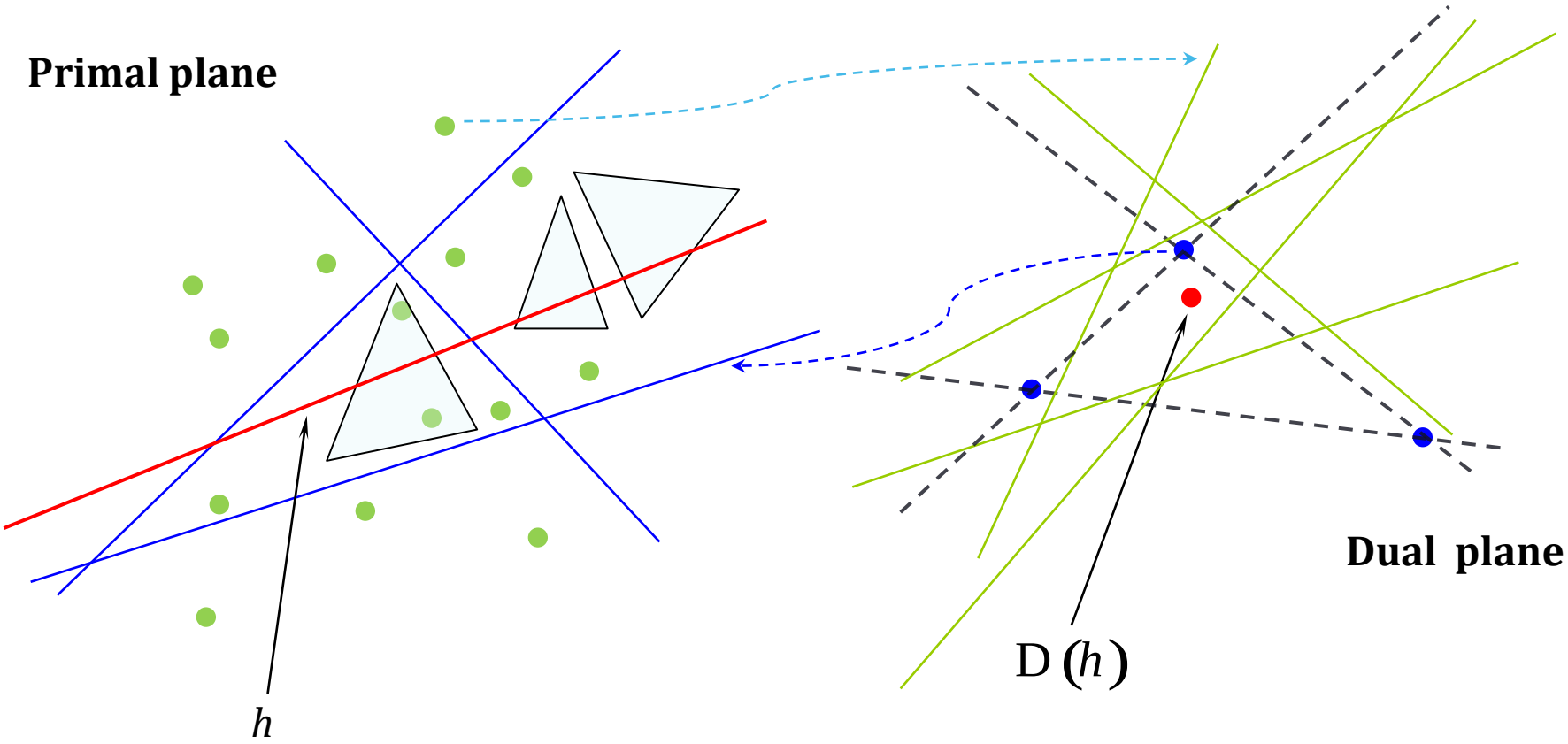


Duality (projekce)

- Aplikujeme m-c duality na množinu bodů S a na přímky dotazovaného trojúhelníku

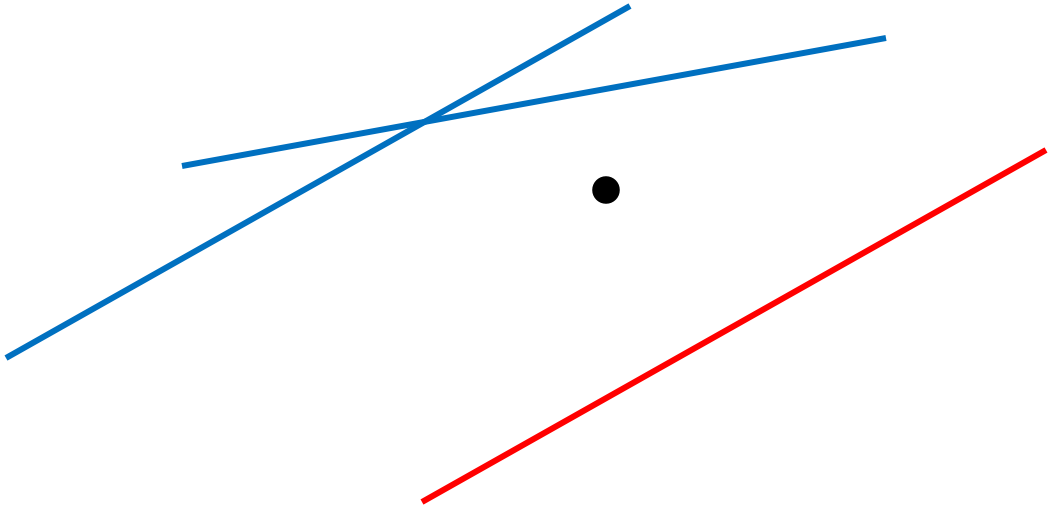
- Množina bodů S se převede na množinu přímek S* a přímky trojúhelníku l1, l2, l3 se převedou na body l1* ,l2* ,l3*

Primal Plane – Dual Plane



Dual plane

- Převedením problému na dual plane vznikne problém: máme množinu L, obsahující n přímek na rovině, spočtěme počet přímek ležících nad dotazovaným bodem.



(1/r)-cutting (řez)

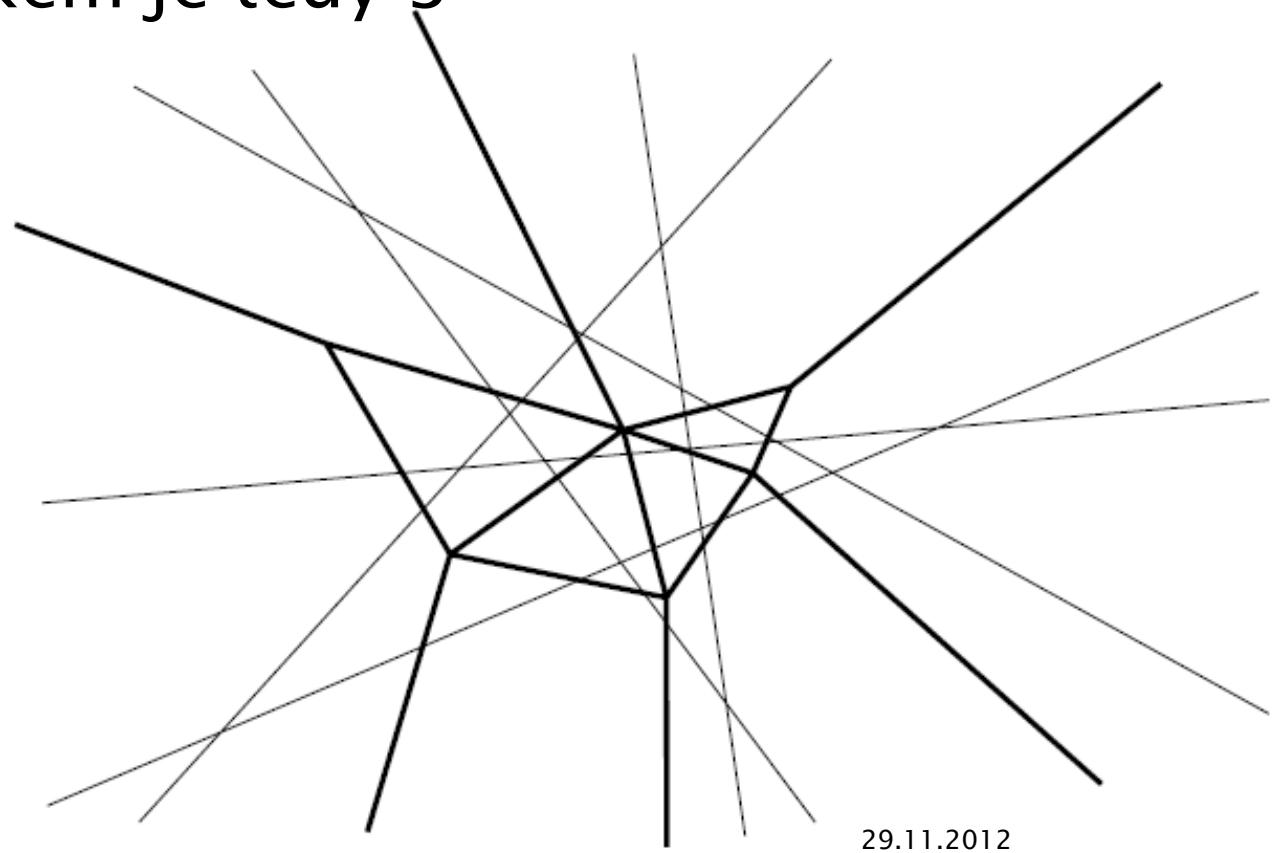
- Cutting lemma – Pro každou množinu L, obsahující n přímek v rovině a parametr r kde $1 \leq r \leq n$, existuje (1/r)-cutting o velikosti O(r^2)

- (1/r)-cutting je rozdělení roviny na trojúhelníky (některé neuzavřené) kde každým trojúhelníkem prochází maximálně n/r přímek.

- Velikost řezu je definován jako počet těchto trojúhelníků

Příklad cutting

- (1/2)-cutting o velikosti 10 v rovině s množinou 6-ti přímek
- n/r=6/2=3, maximum přímek procházejících trojúhelníkem je tedy 3



Cutting

- Proč rozdělovat přímky na trojúhelníky?
  - Nejdříve zjistíme v jakém trojúhelníku se nachází hledaný bod
  - Protože zjišťujeme které přímky leží nad tímto trojúhelníkem, je jasné že tyto přímky leží také nad tímto bodem.
  - Opačně, když přímky leží pod trojúhelníkem leží také pod hledaným bodem
  - O jediných přímkách nevíme jestli jsou nad nebo pod bodem a to u přímek procházejících trojúhelníkem ve kterém leží hledaný bod

Cutting

- Vezmu tedy přímky, které procházely trojúhelníkem a znovu je rozdělím pomocí (1/r)-cutting
- Vzniknou mi tedy další trojúhelníky na menším počtu přímek kde provádím stejné operace jako na celkové množině přímek
- Je to vlastně rekurzivní provádění na menších a menších částech

Struktura cutting trees

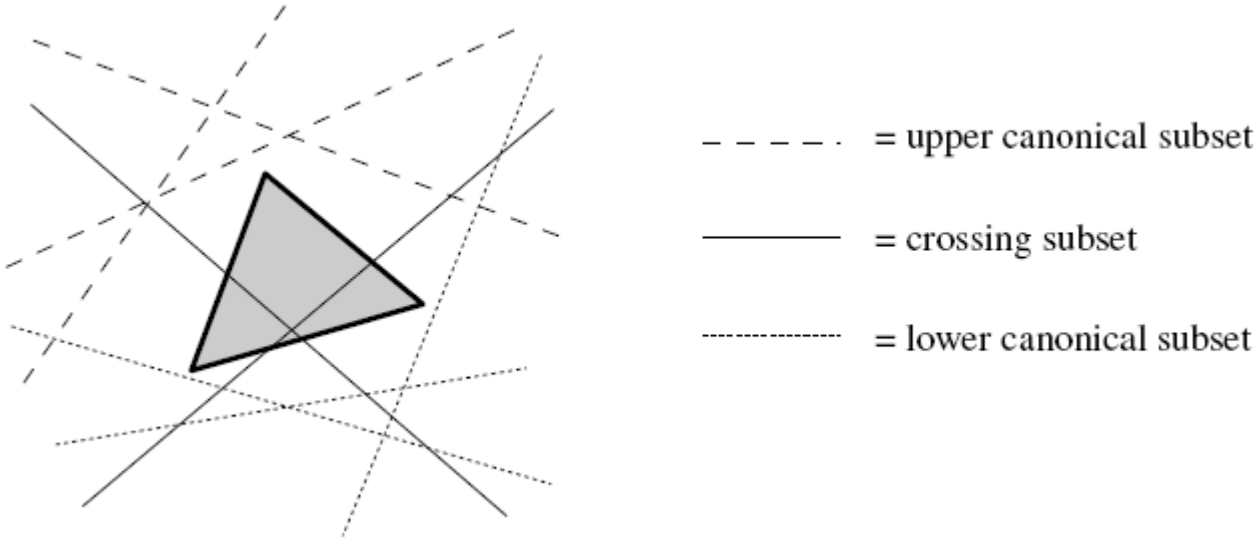
- Když množina L obsahuje pouze jednu přímku tak se cutting tree skládá z pouze jednoho listu kde je množina L explicitně uložena.

Struktura cutting trees

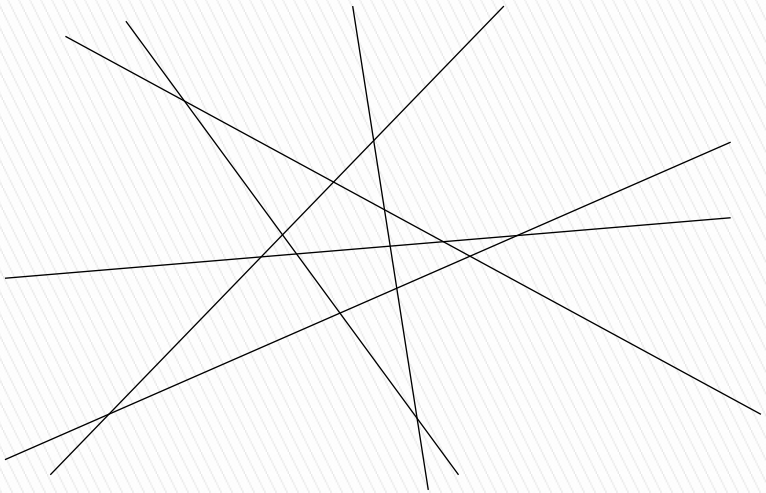
- Jestliže množina L má více než jednu přímku je struktura strom.
- Strom se skládá z kořene a potomků, kde každý potomek (v) koresponduje s trojúhelníkem z (1/r)-cutting to značíme t(v).
- Podmnožina přímek:
  - které leží pod t(v) říkáme nižší kanonický podmnožina a značíme L−(v)
  - které leží nad t(v) říkáme vyšší kanonický podmnožina a značíme L+(v)
  - které protínají t(v) říkáme protínající podmnožina

Struktura cutting trees

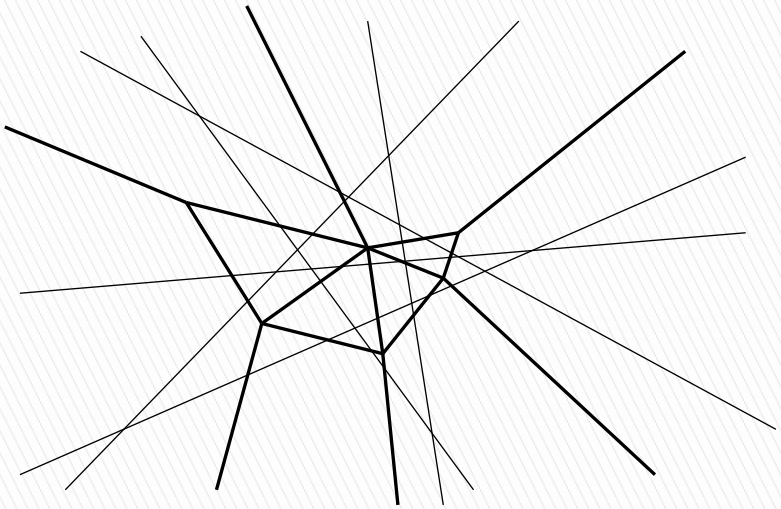
- Kanonické podmnožiny a protínající podmnožina trojúhelníku



Konstrukce cutting trees

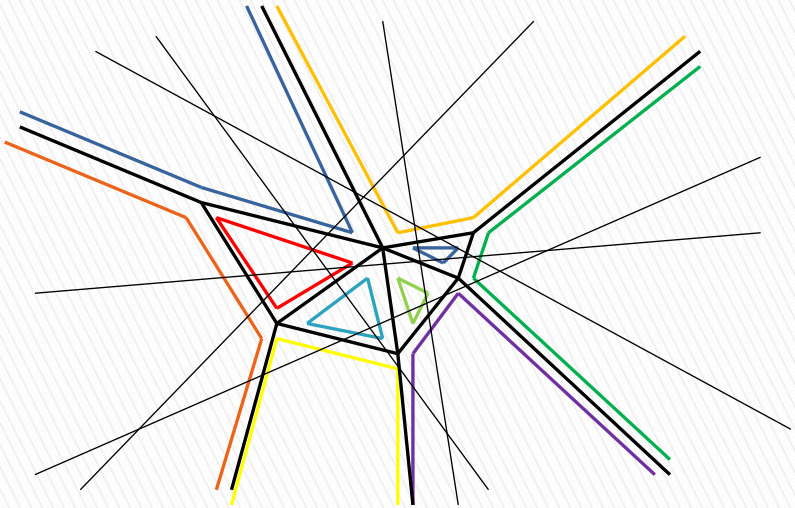


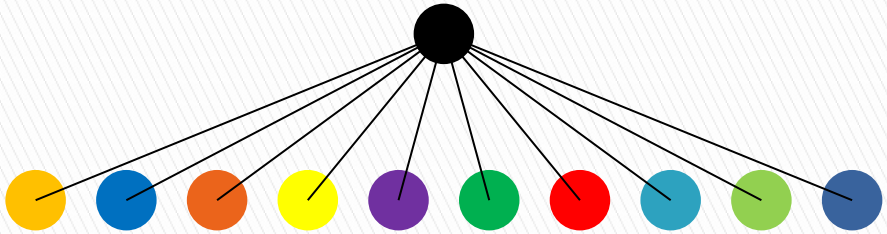
Konstrukce cutting trees



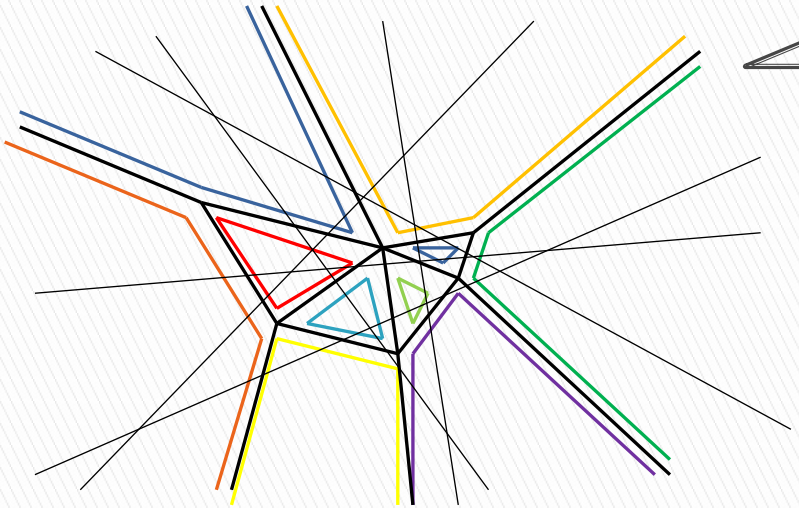
Cutting lemma!

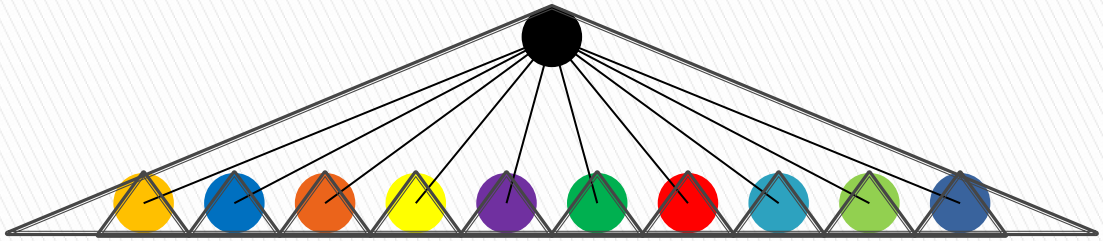
Konstrukce cutting trees



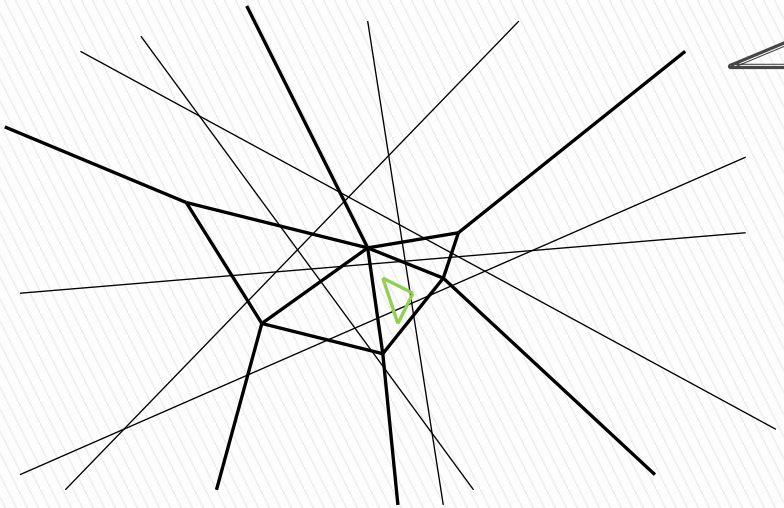


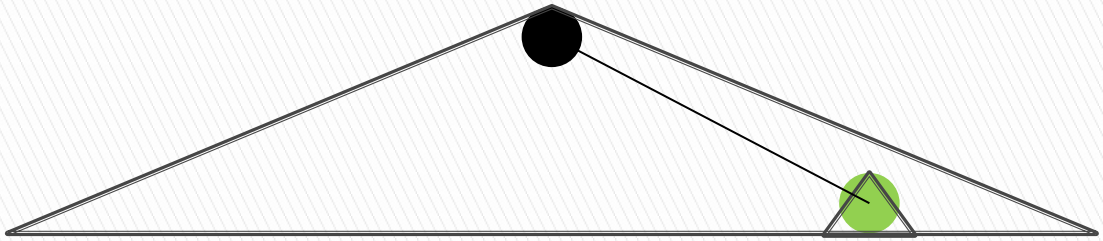
Konstrukce cutting trees



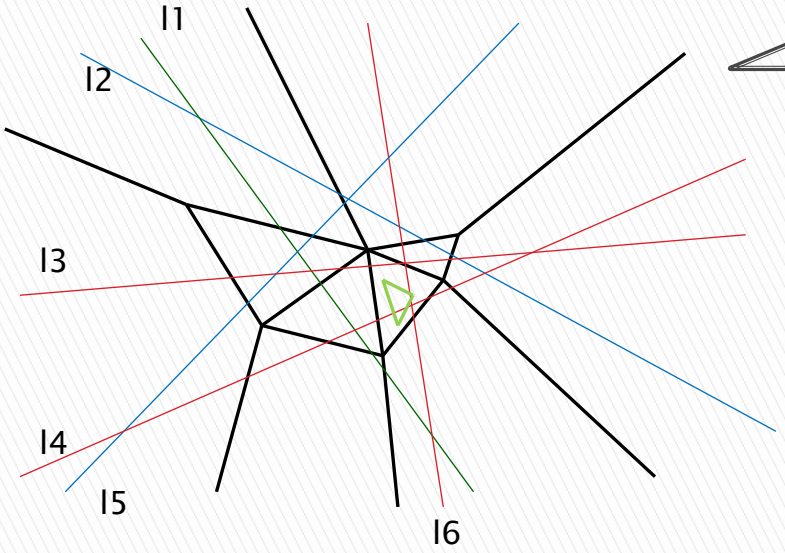


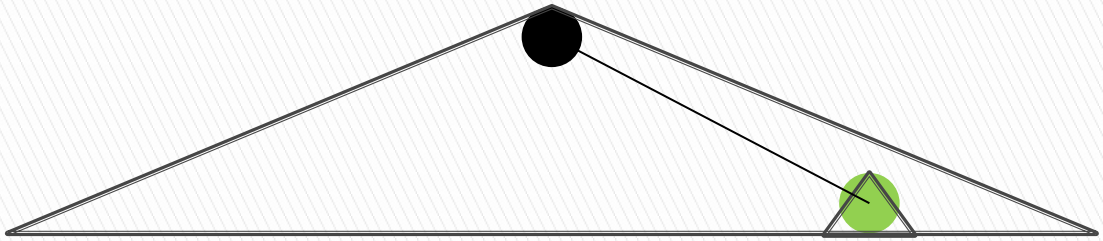
Konstrukce cutting trees





Konstrukce cutting trees



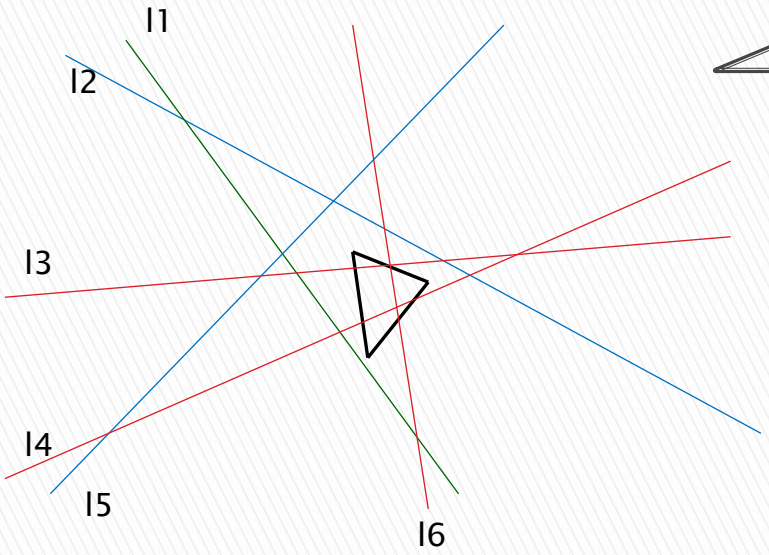


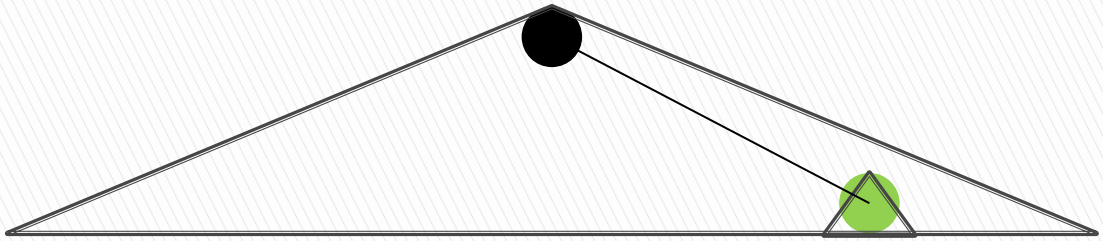
L−(v) = l1

L+(v) = l2, l5

protínající podmnožina = l3, l4, l6

Konstrukce cutting trees



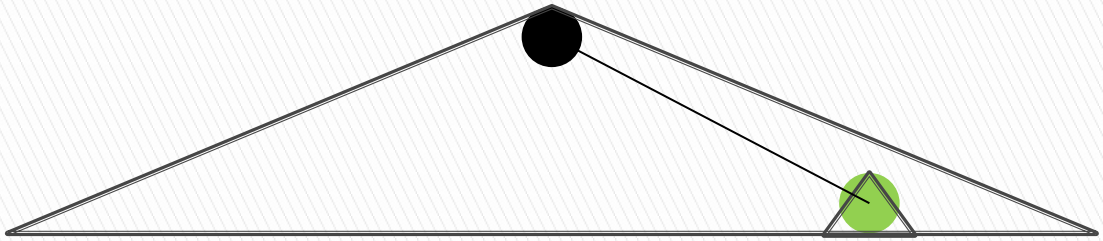


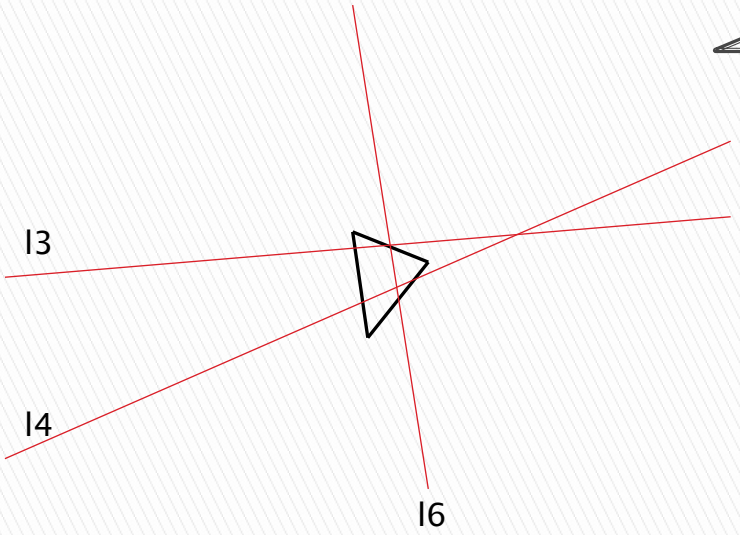
L−(v) = l1

L+(v) = l2, l5

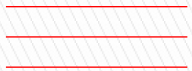
protínající podmnožina = l3, l4, l6

Konstrukce cutting trees

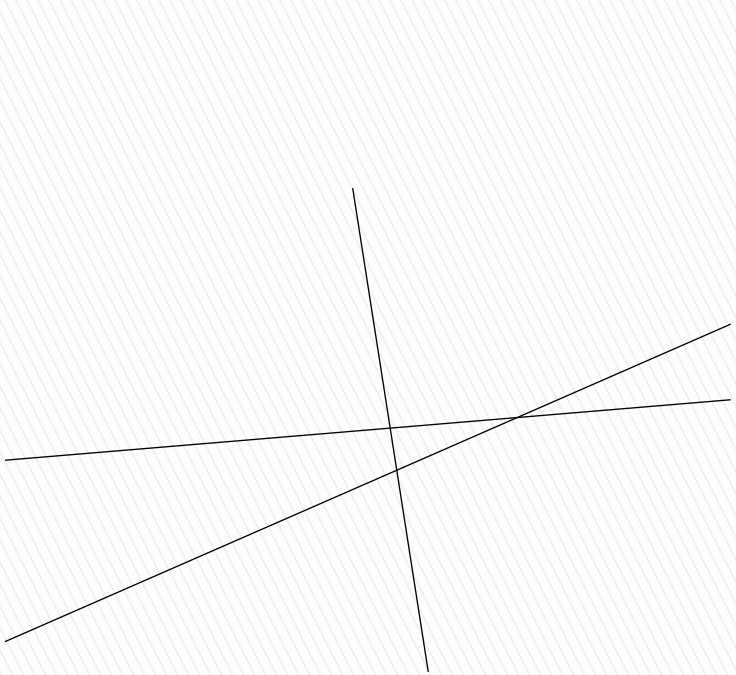


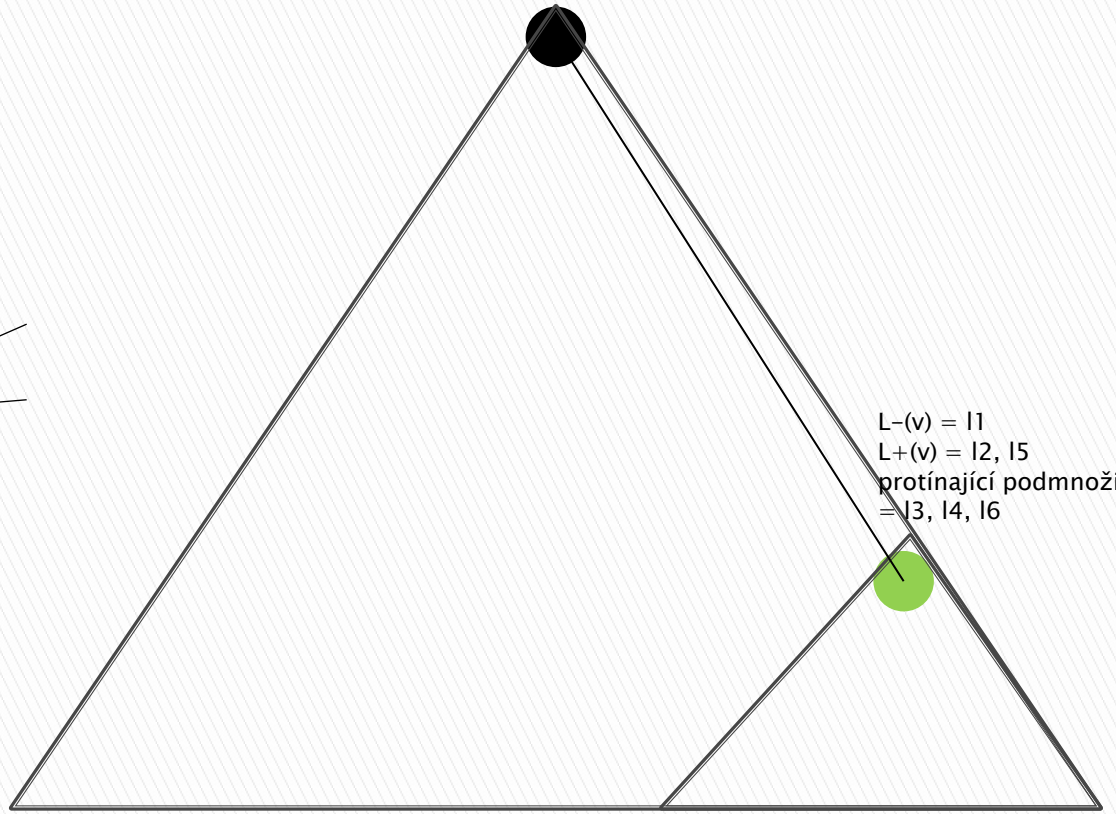


protínající podmnožina = l3, l4, l6

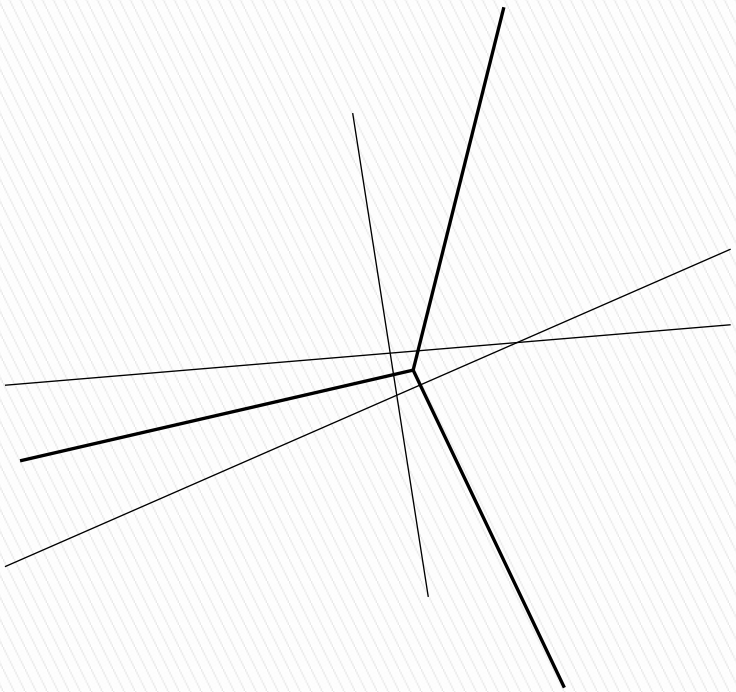


Konstrukce cutting trees

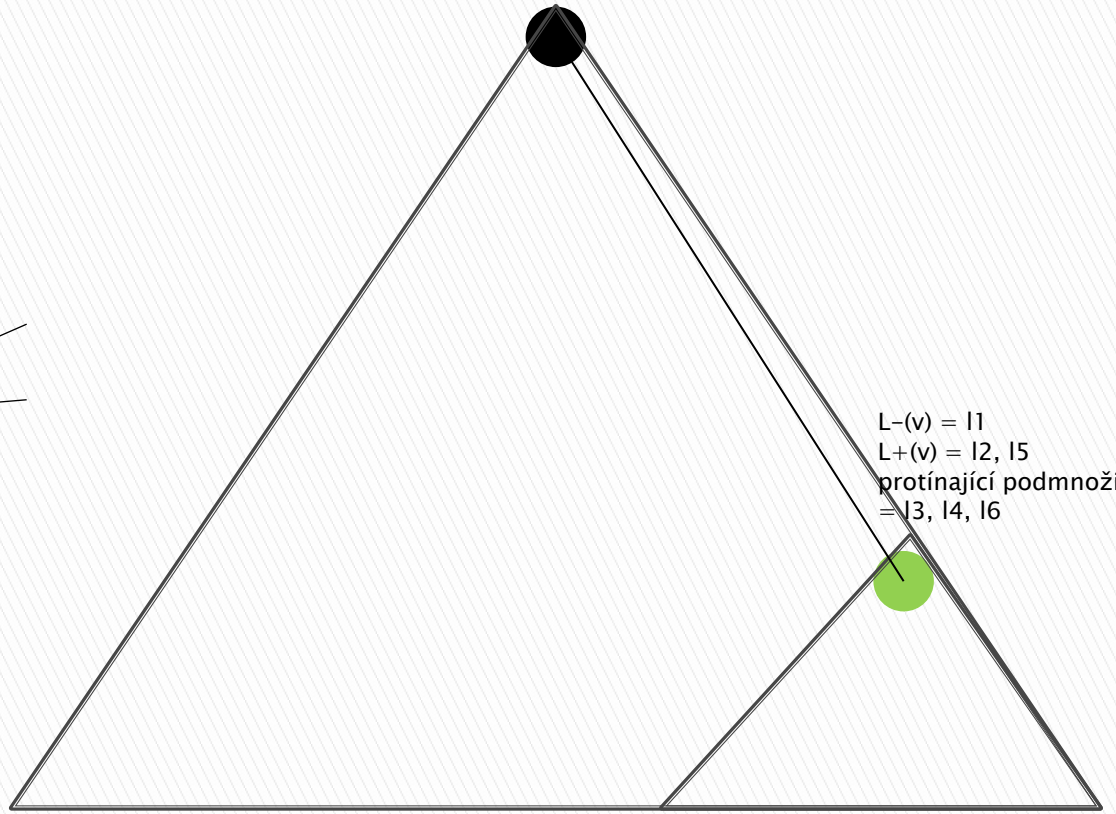




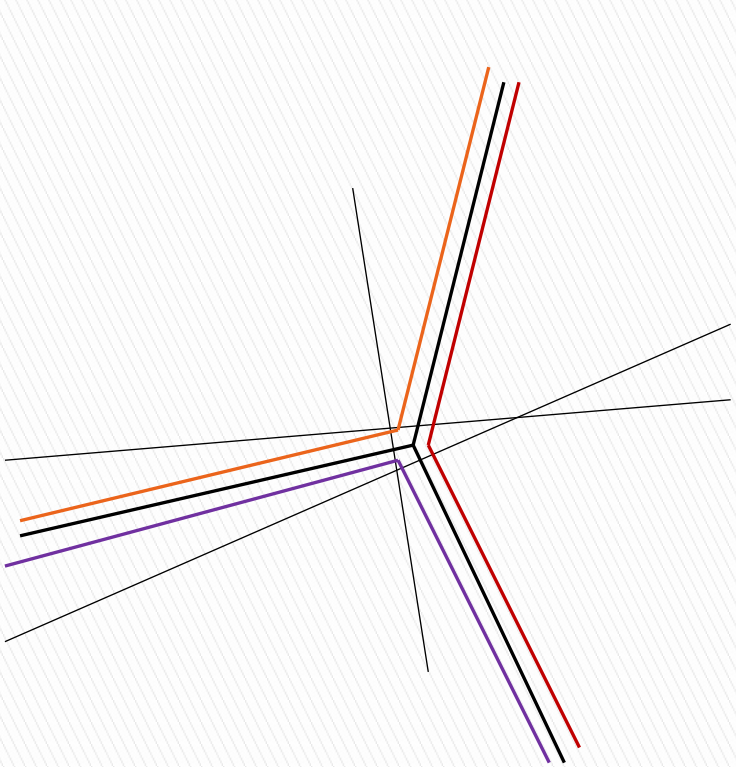
Konstrukce cutting trees

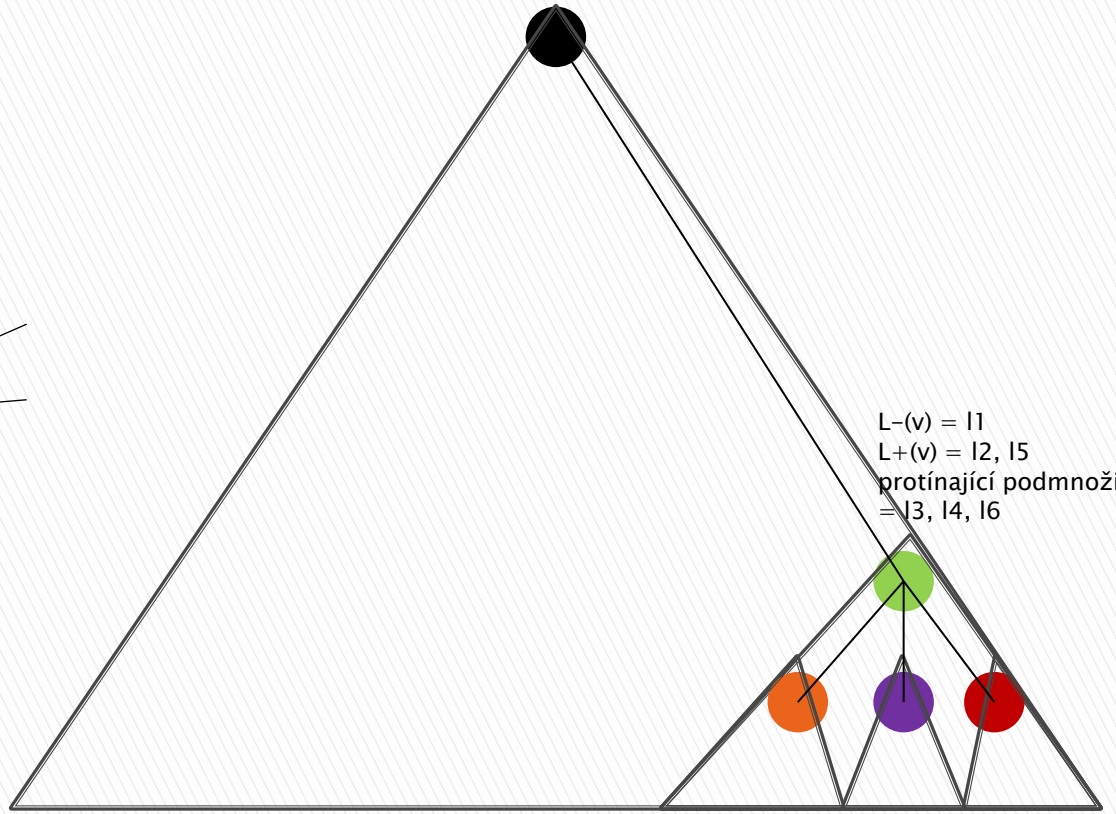


Cutting lemma!

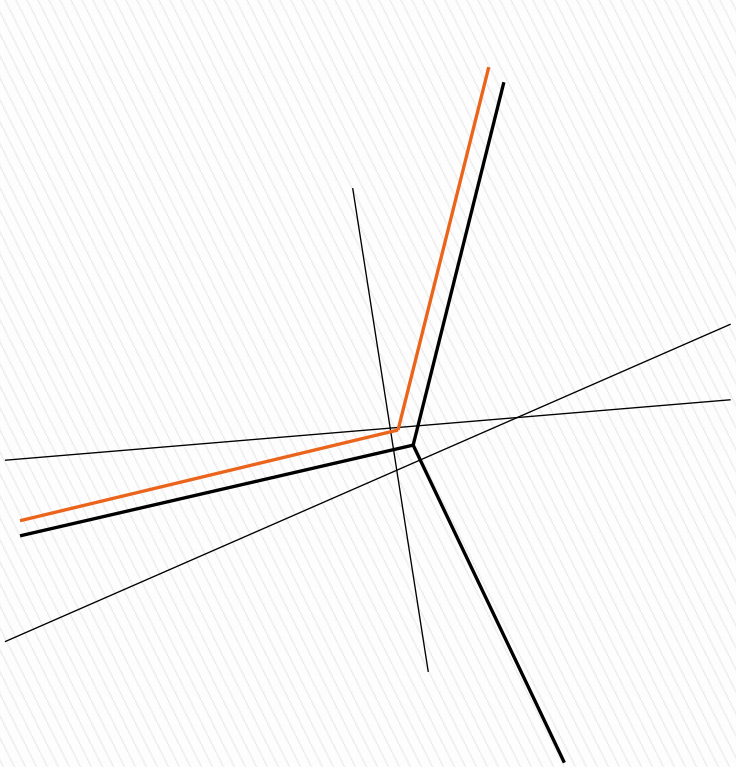


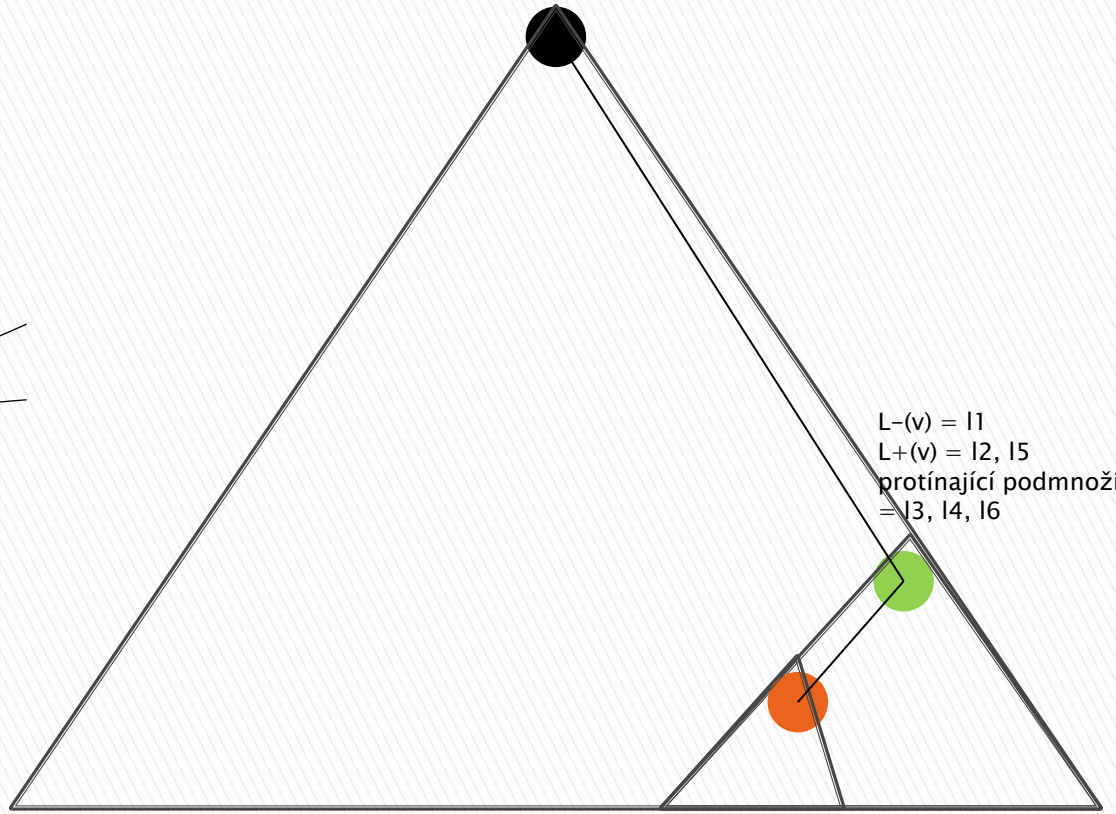
Konstrukce cutting trees



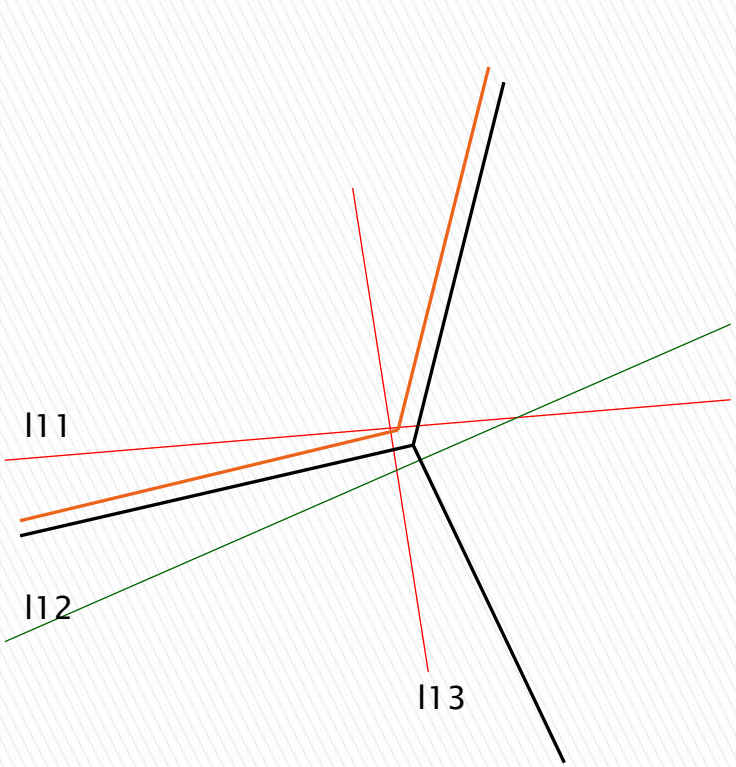


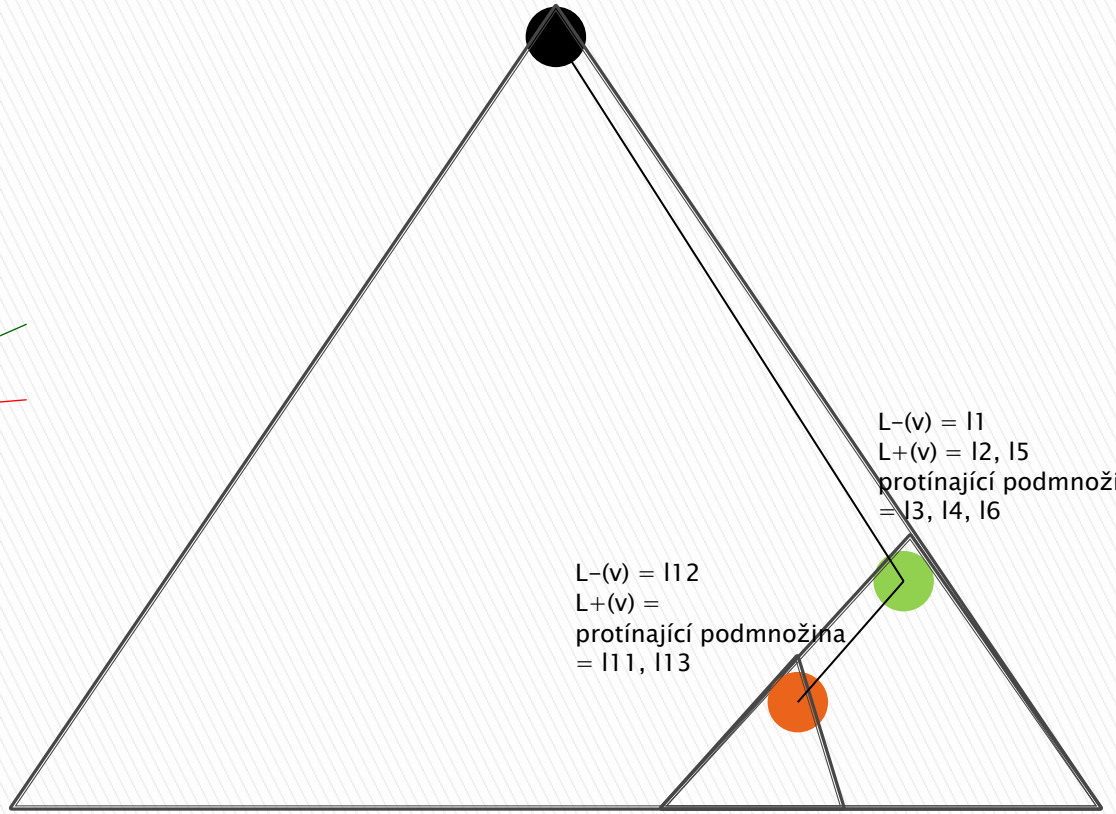
Konstrukce cutting trees



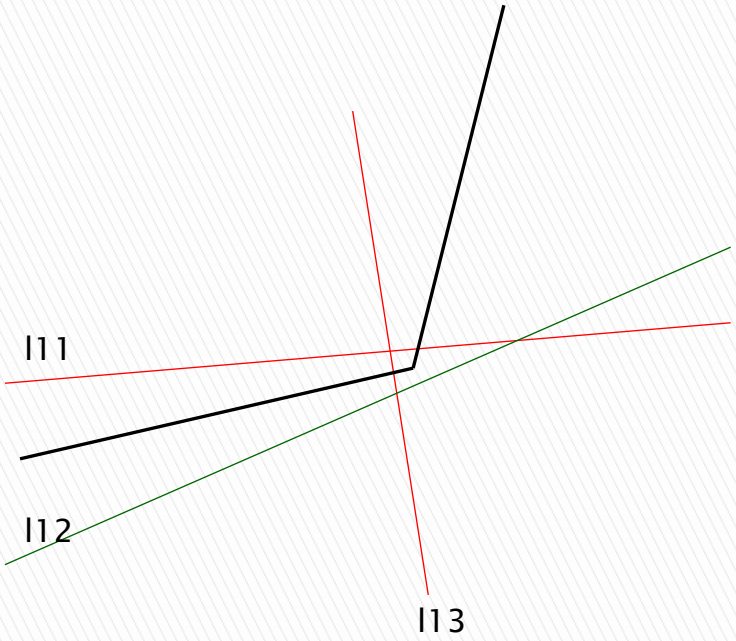


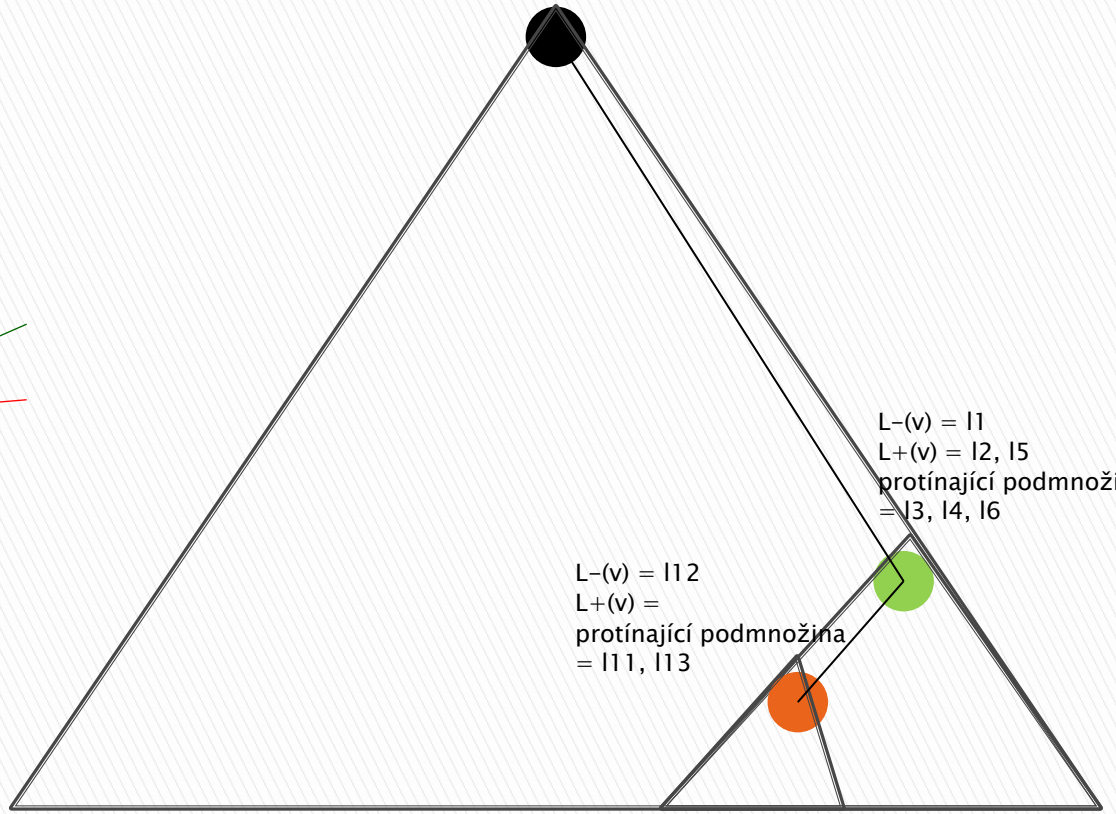
Konstrukce cutting trees



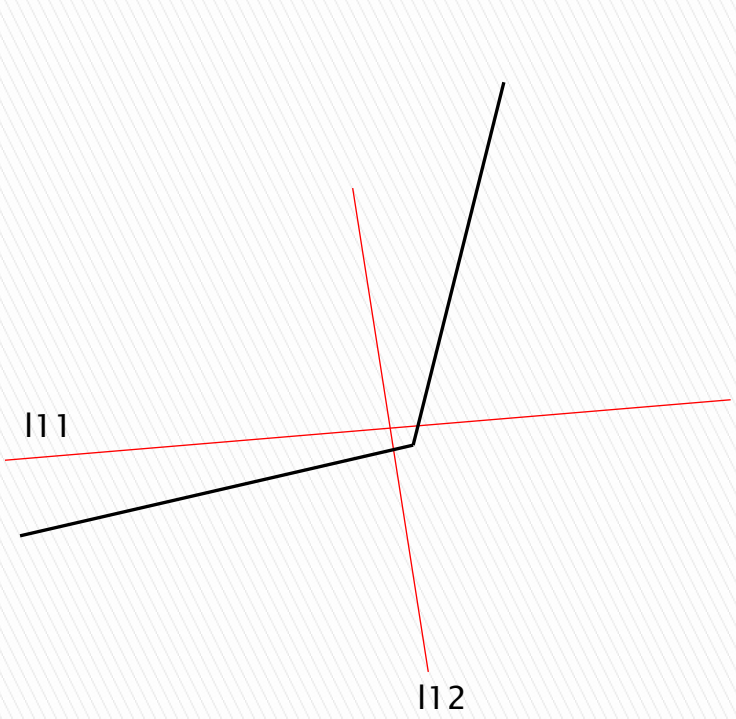


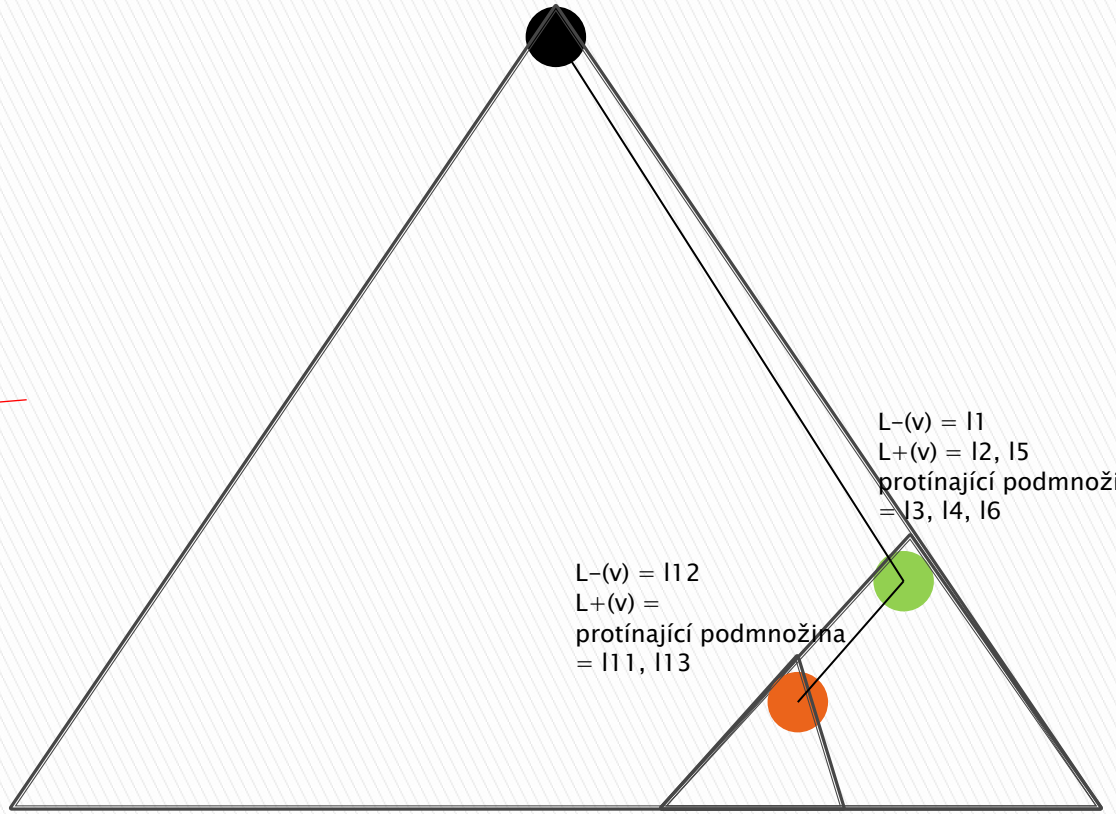
Konstrukce cutting trees



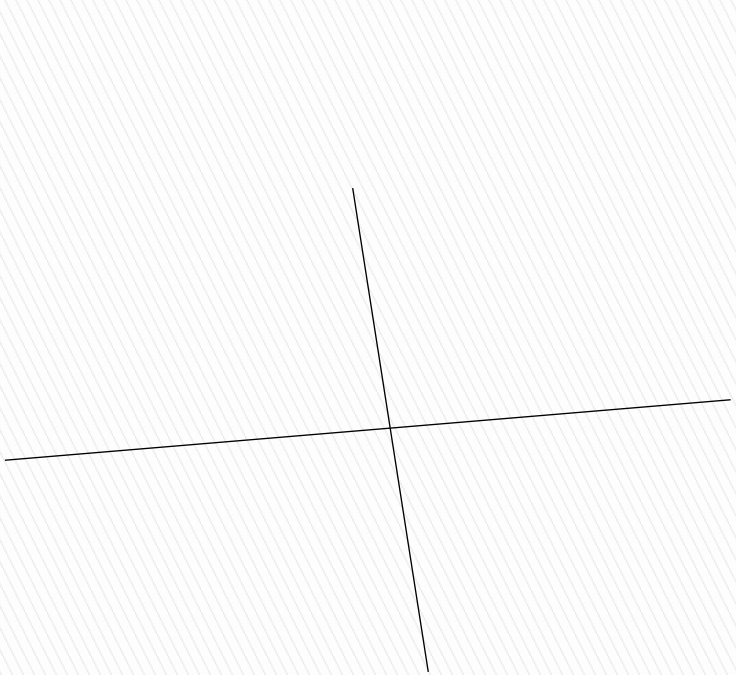


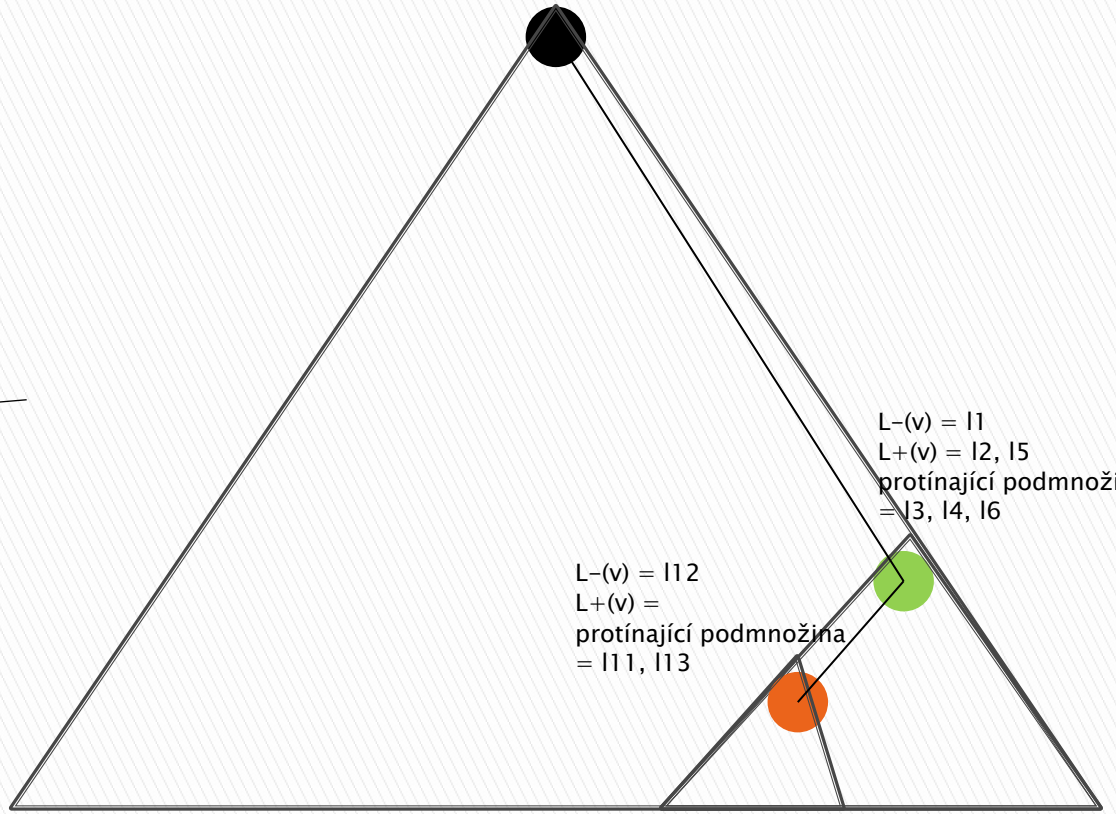
Konstrukce cutting trees





Konstrukce cutting trees



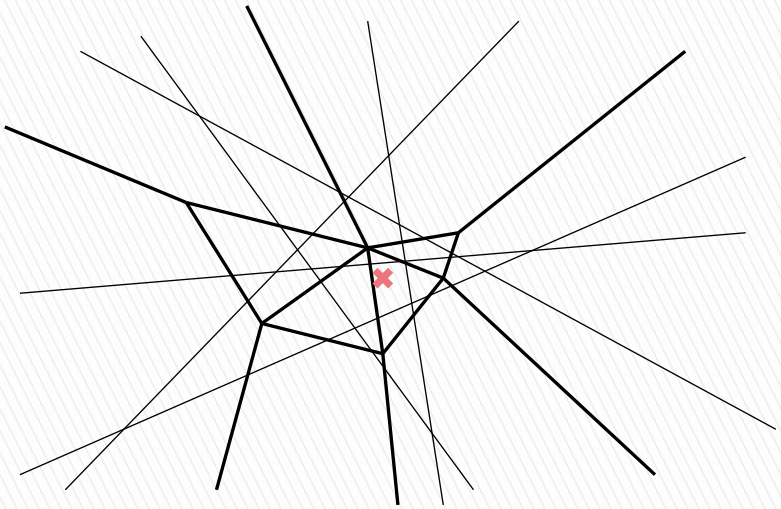


Použití cutting trees

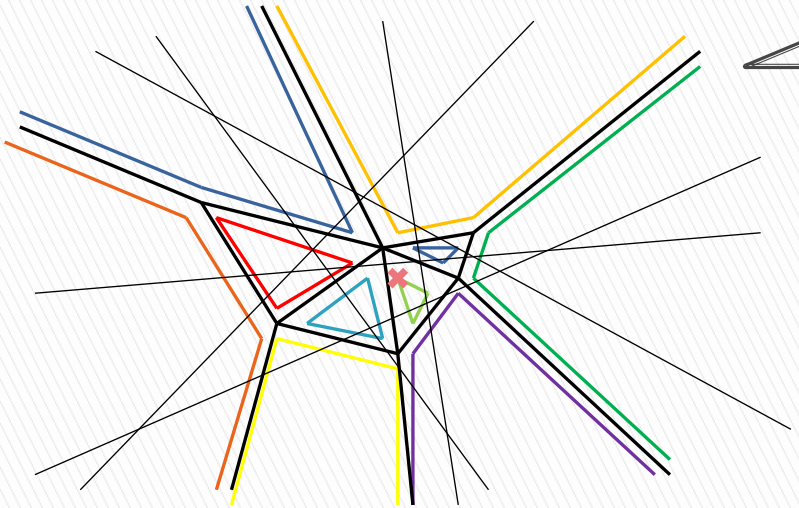
Algorithm SELECTBELOWPOINT$(q, \mathcal{T})$
Input. A query point q and a cutting tree or subtree of it.
Output. A set of canonical nodes for all lines in the tree that lie below q.

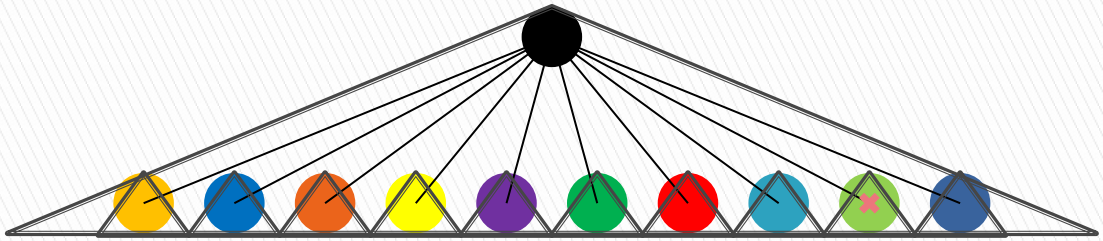
1. $\Upsilon \leftarrow \emptyset$
2. **if** $\mathcal{T}$ consists of a single leaf μ
3. **then if** the line stored at μ lies below q **then** $\Upsilon \leftarrow \{\mu\}$
4. **else for** each child v of the root of $\mathcal{T}$
5. **do** Check if q lies in $t(v)$.
6. Let v_q be the child such that $q \in t(v_q)$.
7. $\Upsilon \leftarrow \{v_q\} \cup$ SELECTBELOWPOINT$(q, \mathcal{T}_{v_q})$
8. **return** Υ

Konstrukce cutting trees

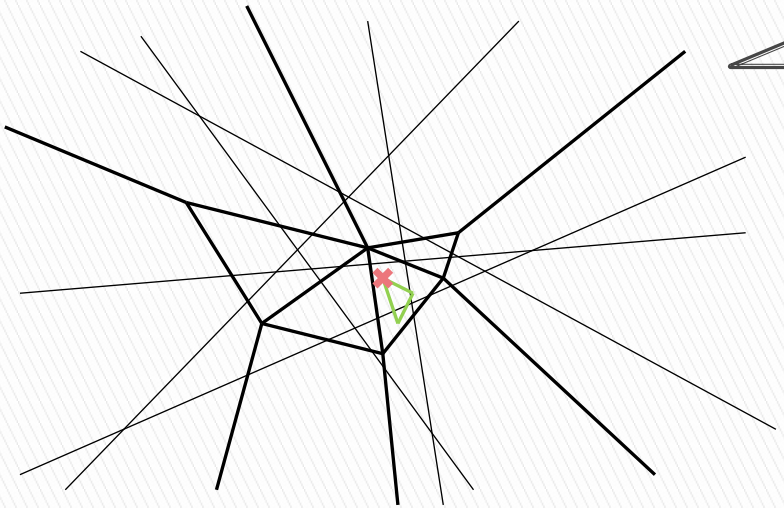


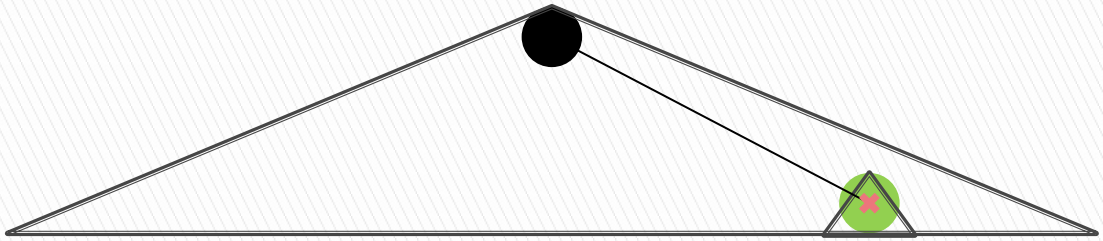
Konstrukce cutting trees



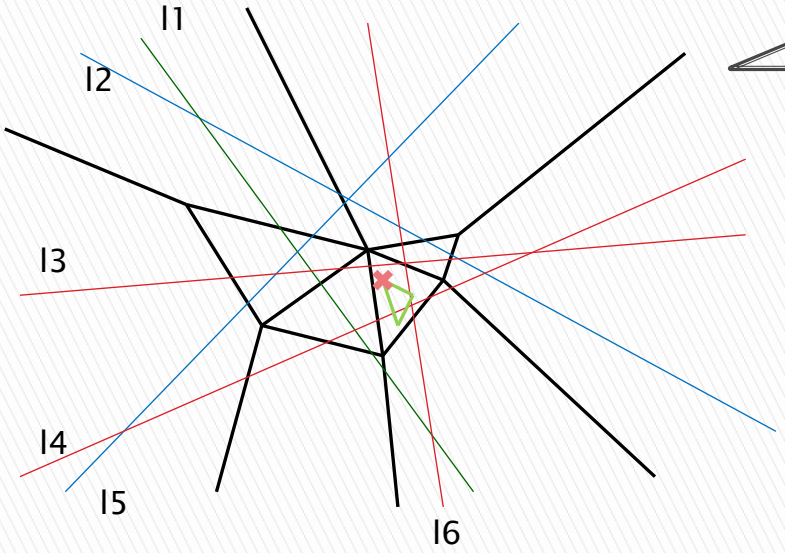


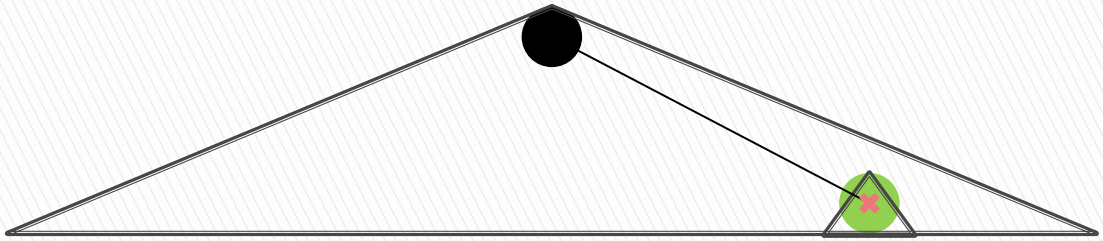
Konstrukce cutting trees





Konstrukce cutting trees



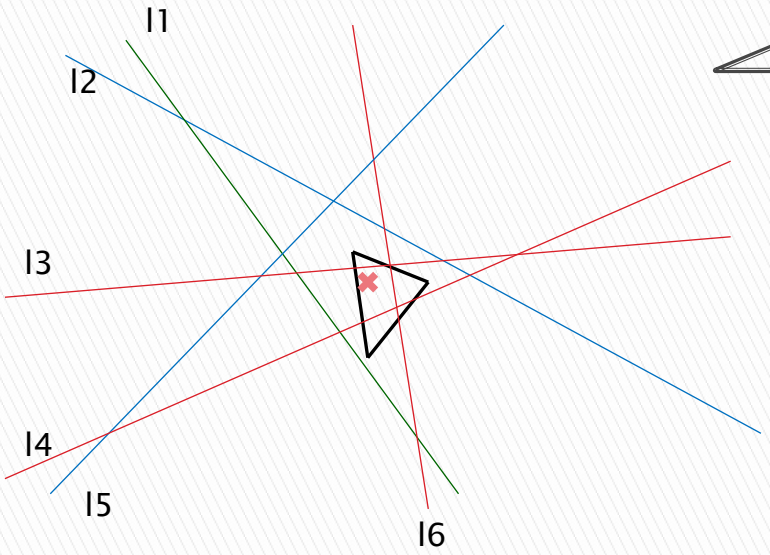


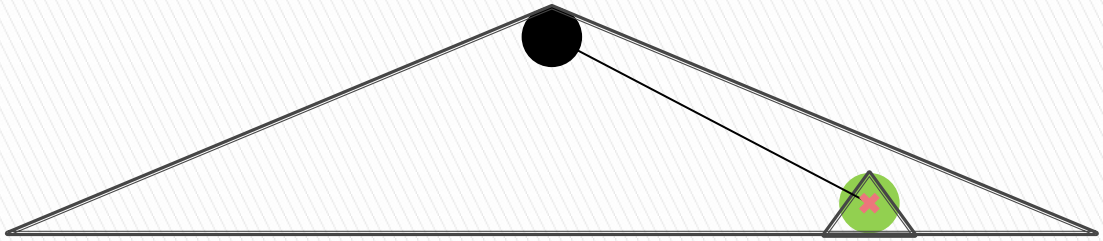
L−(v) = l1

L+(v) = l2, l5

protínající podmnožina = l3, l4, l6

Konstrukce cutting trees



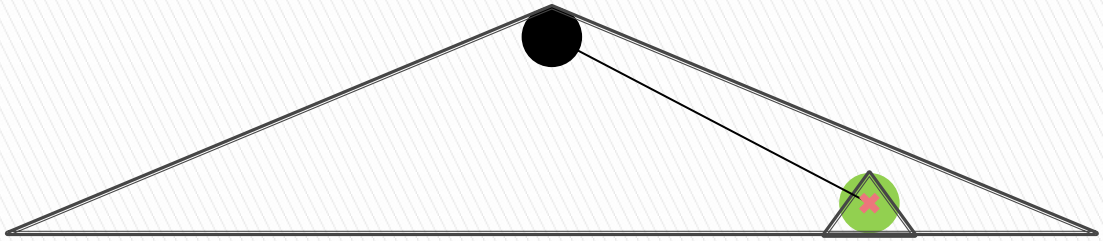


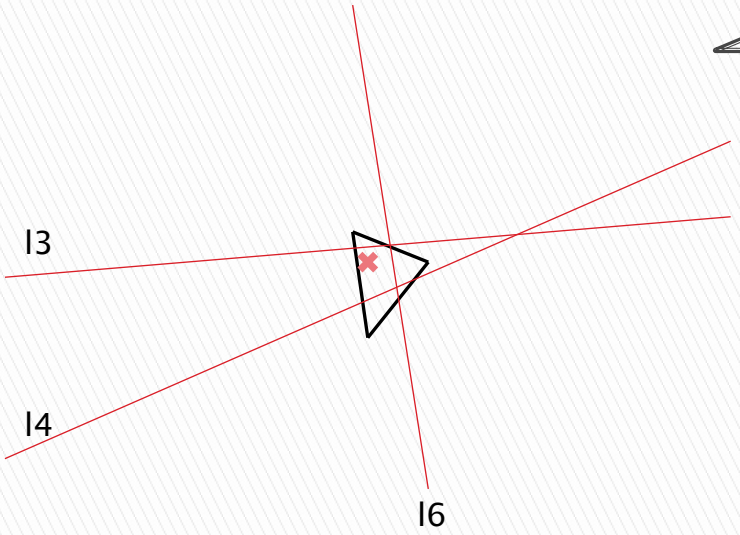
L−(v) = l1

L+(v) = l2, l5

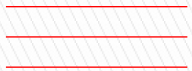
protínající podmnožina = l3, l4, l6

Konstrukce cutting trees

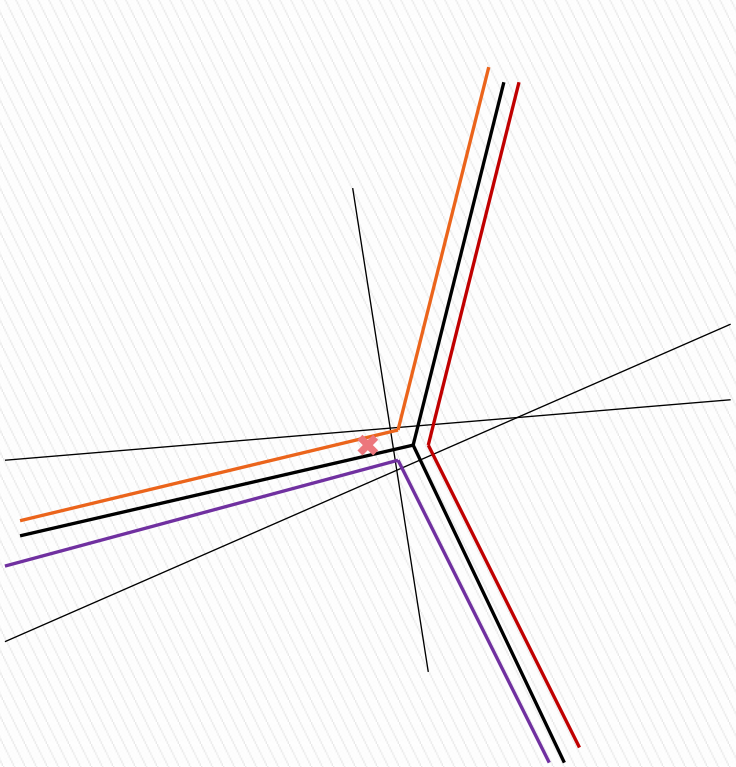


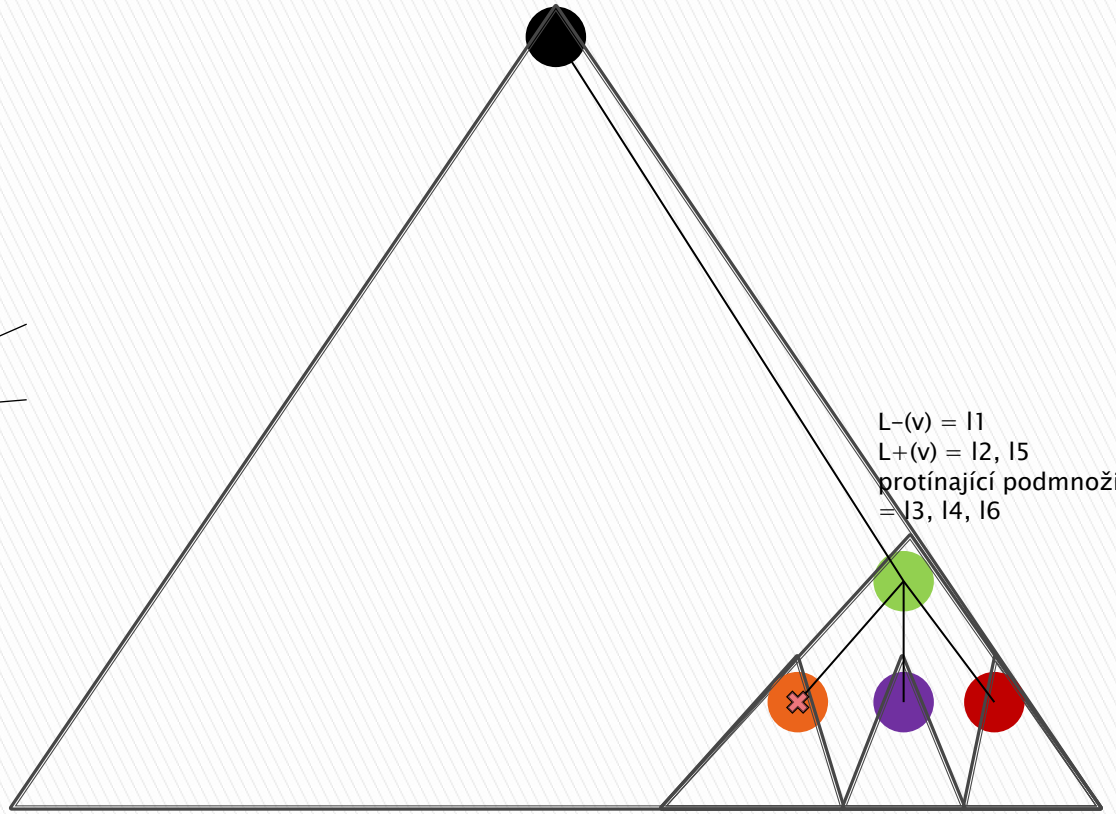


protínající podmnožina = l3, l4, l6

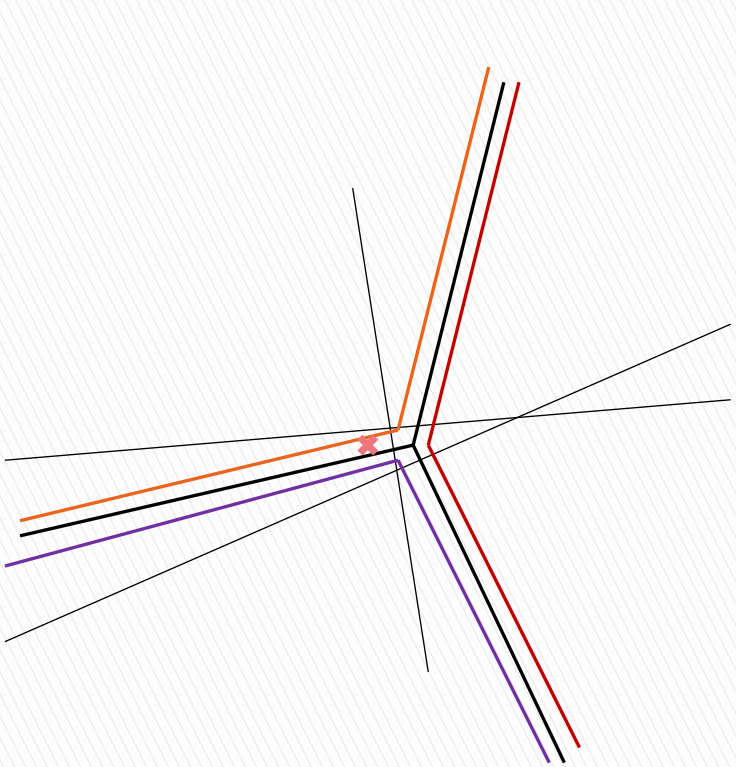


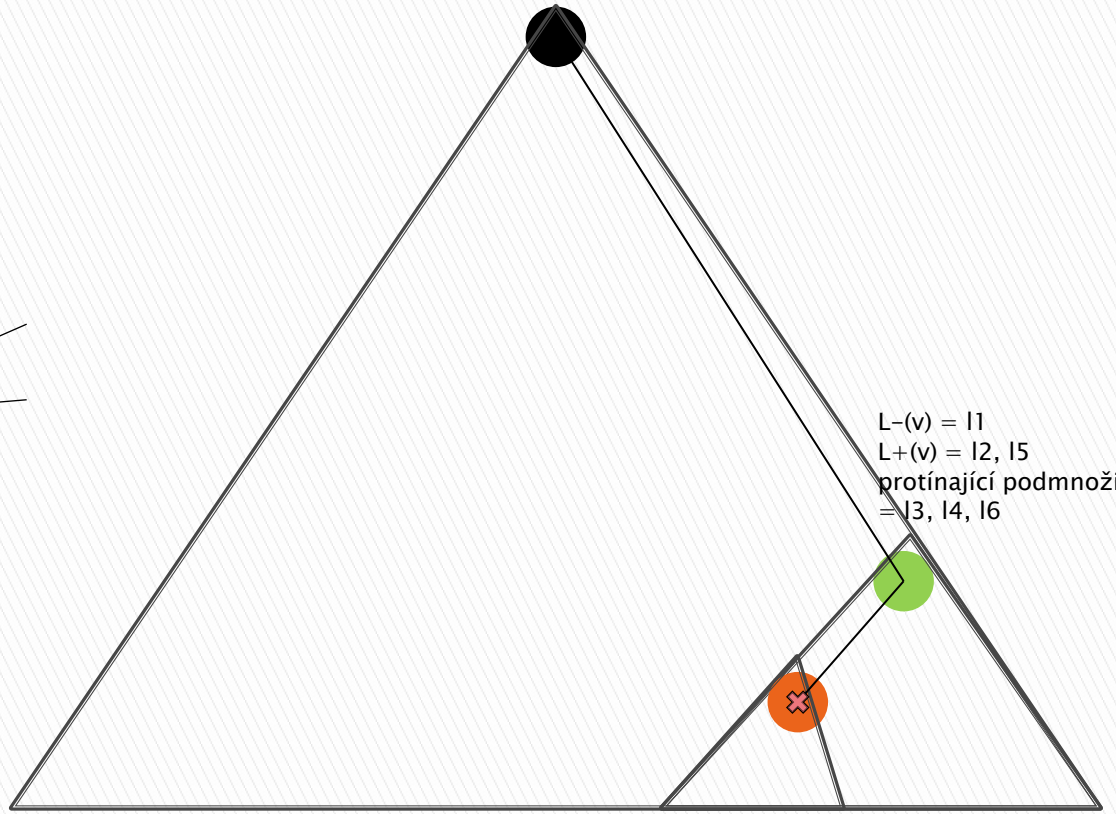
Konstrukce cutting trees



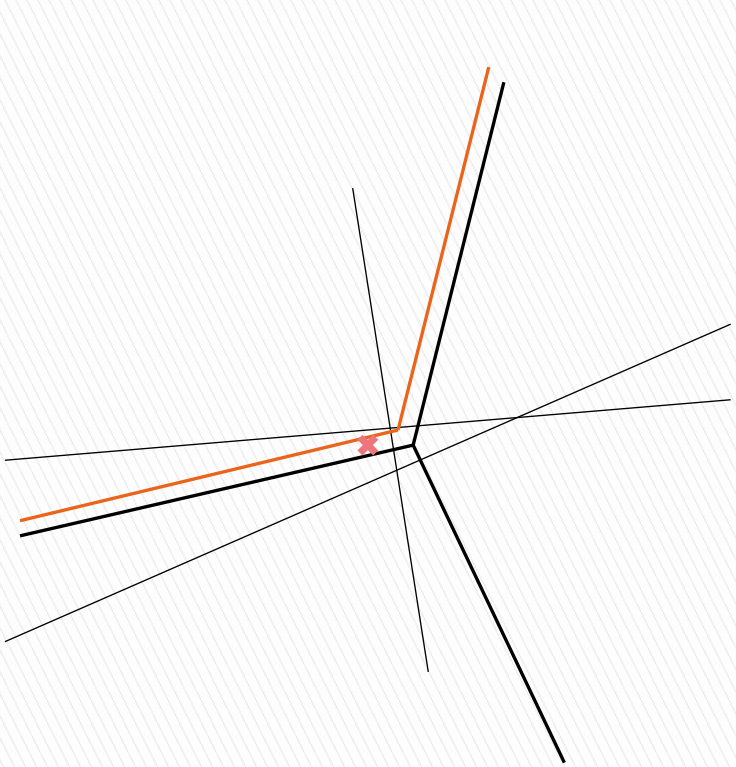


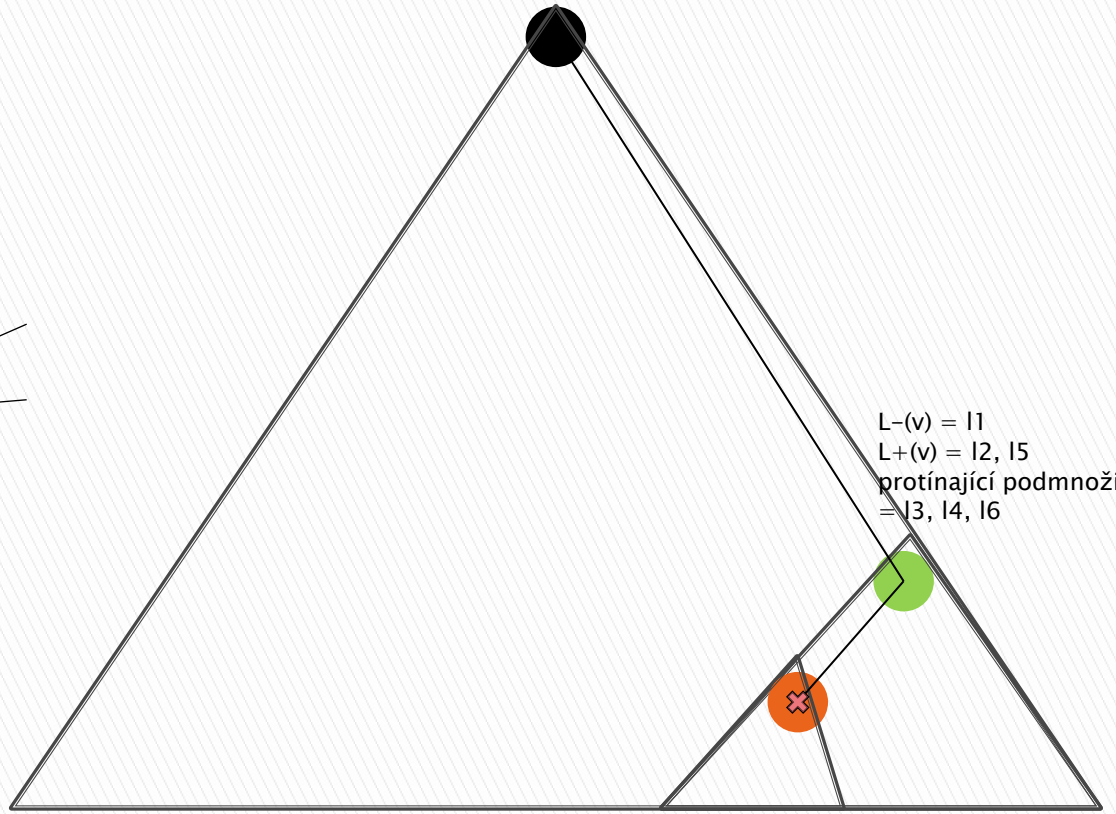
Konstrukce cutting trees



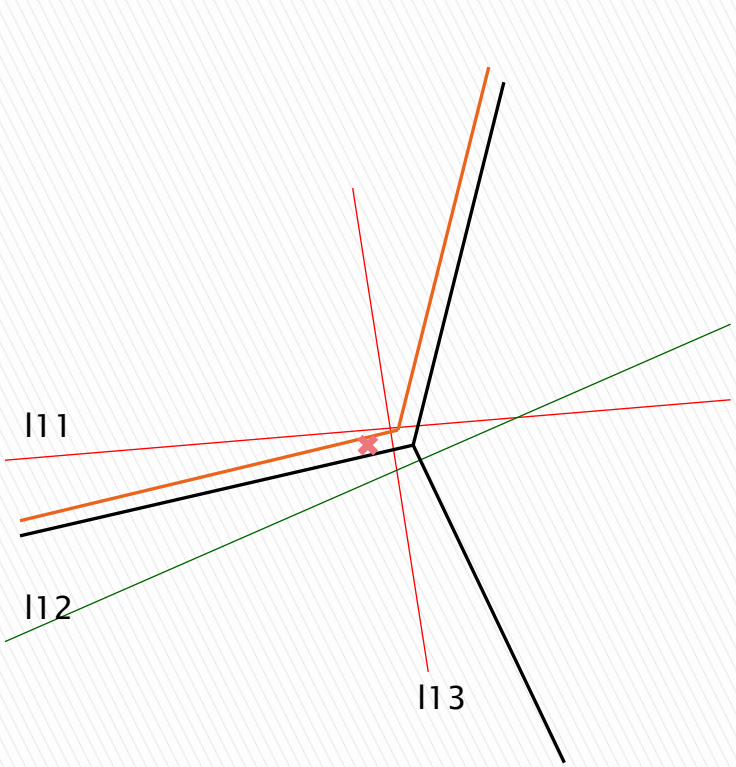


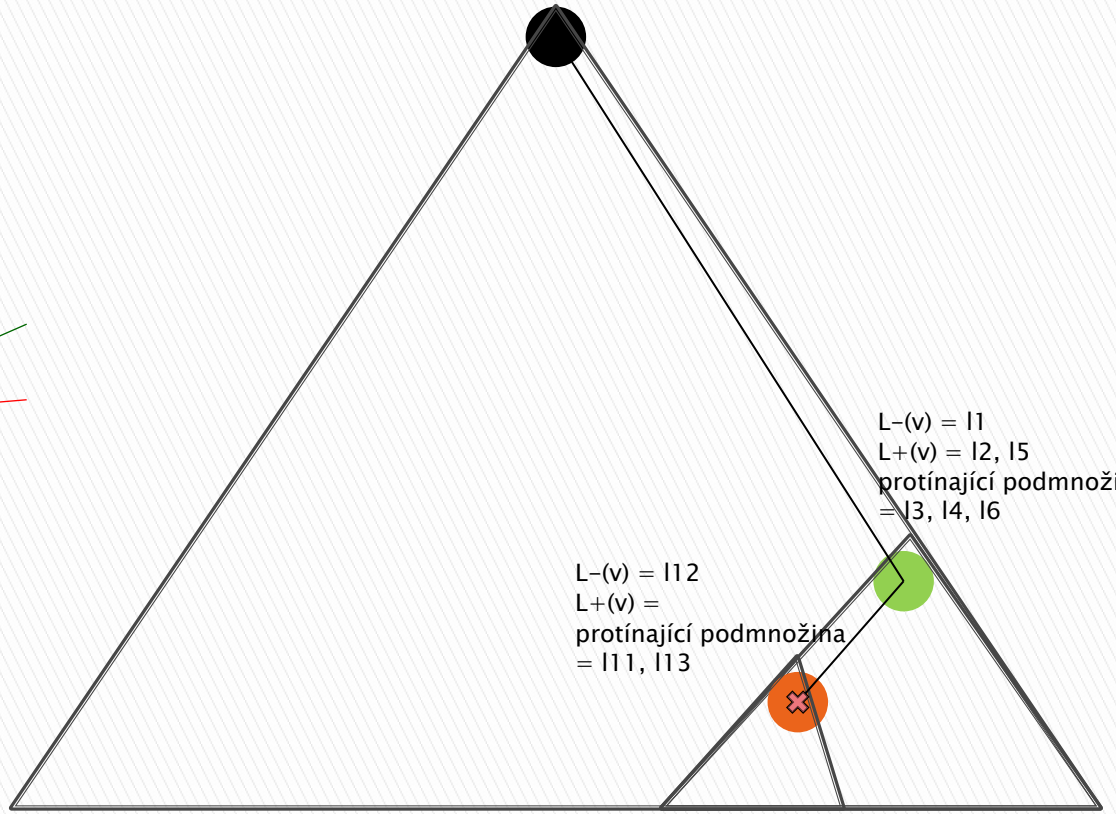
Konstrukce cutting trees



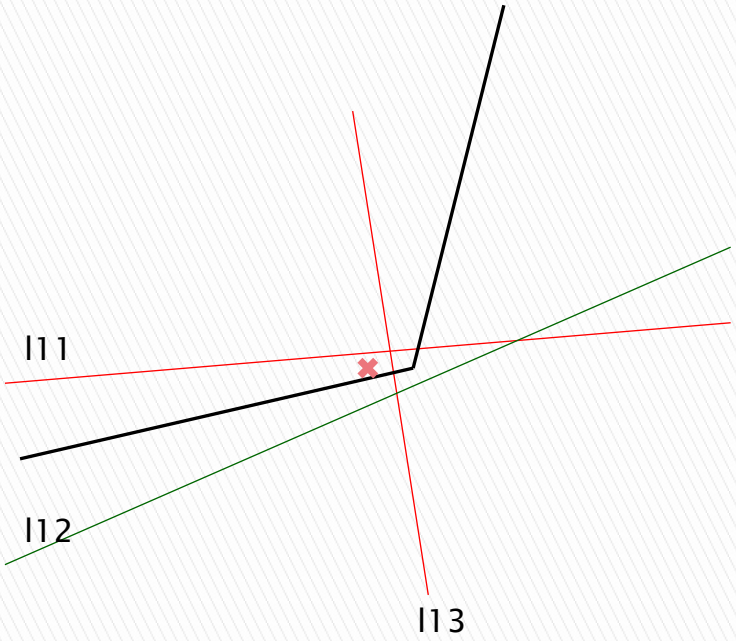


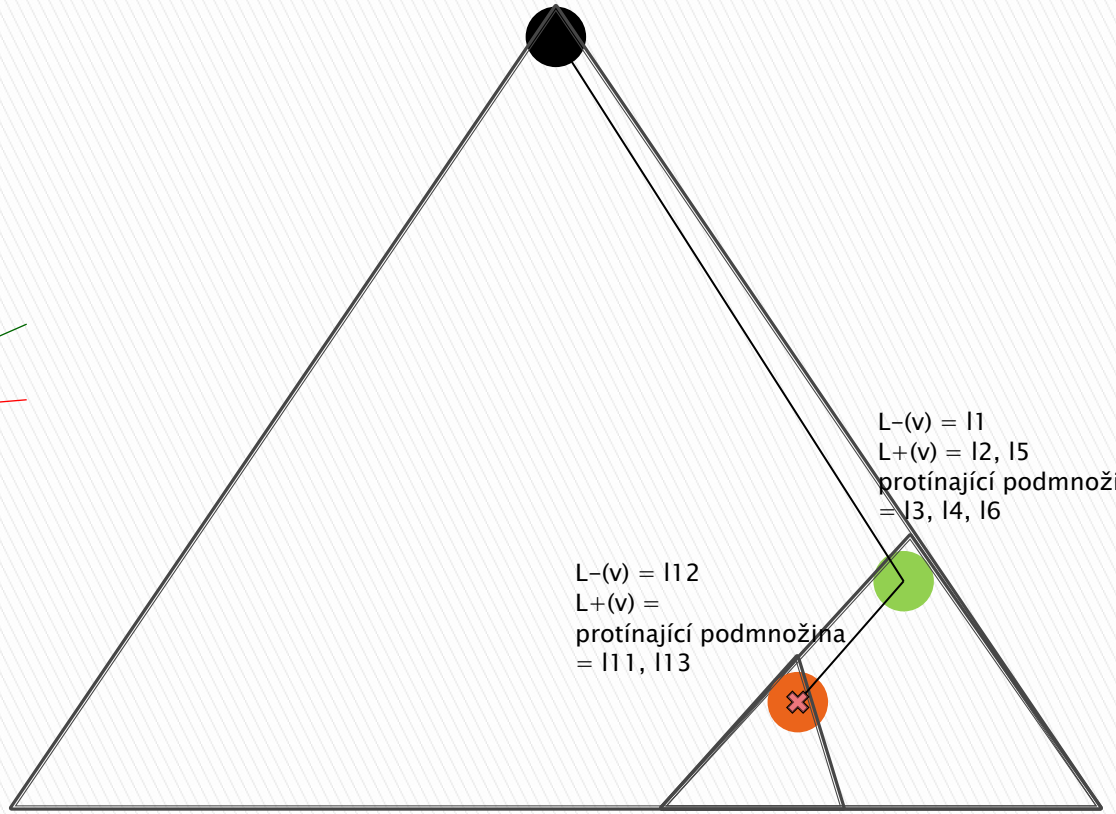
Konstrukce cutting trees



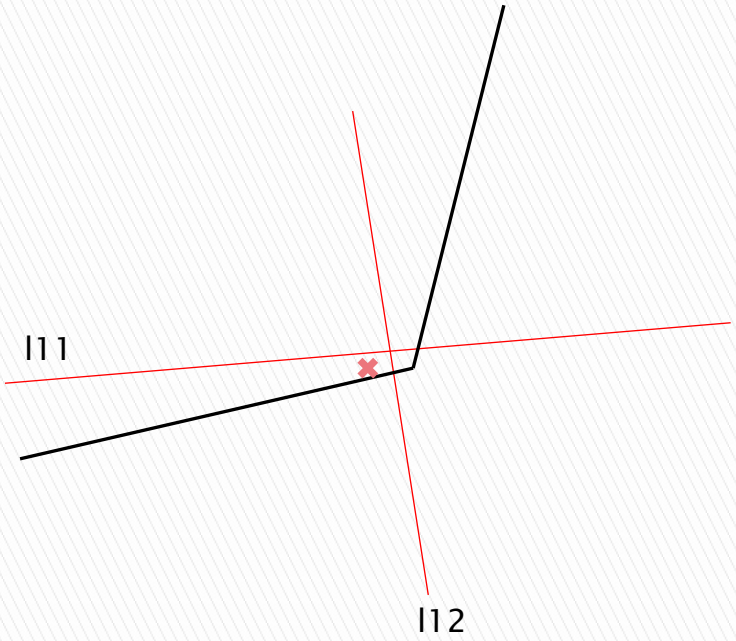


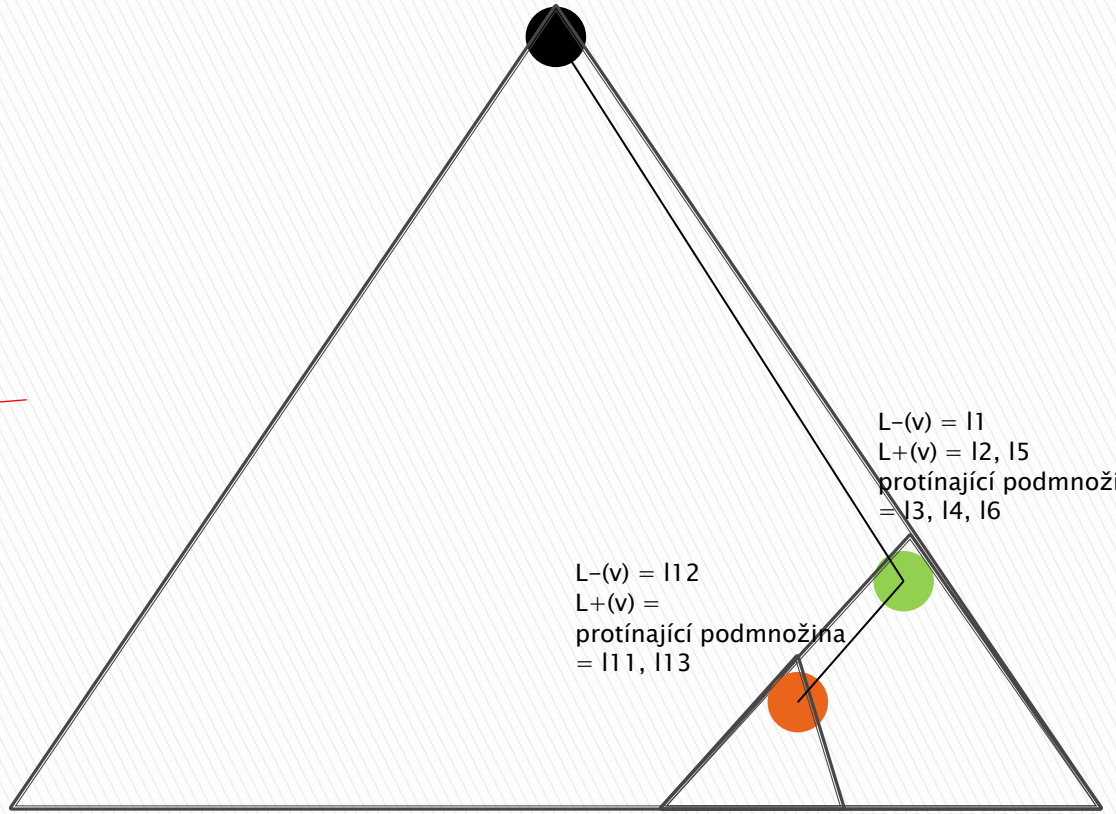
Konstrukce cutting trees



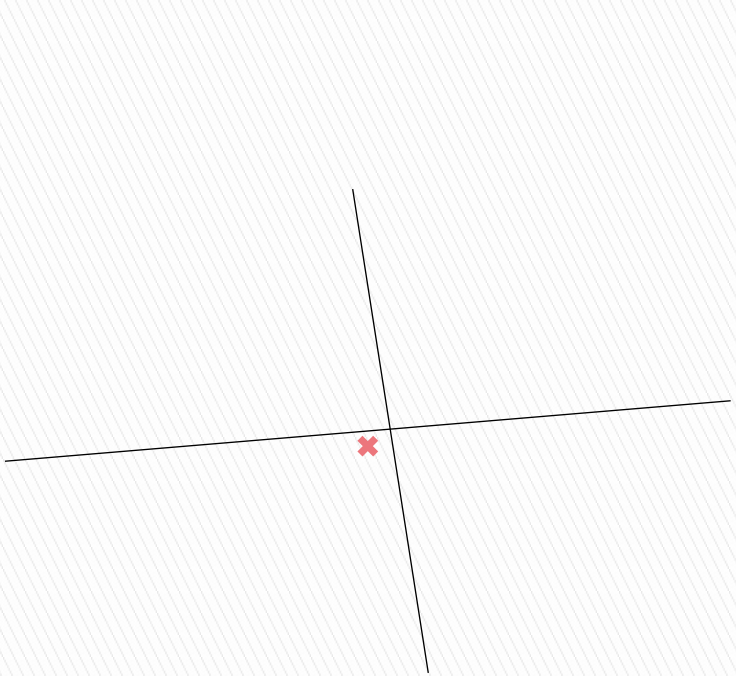


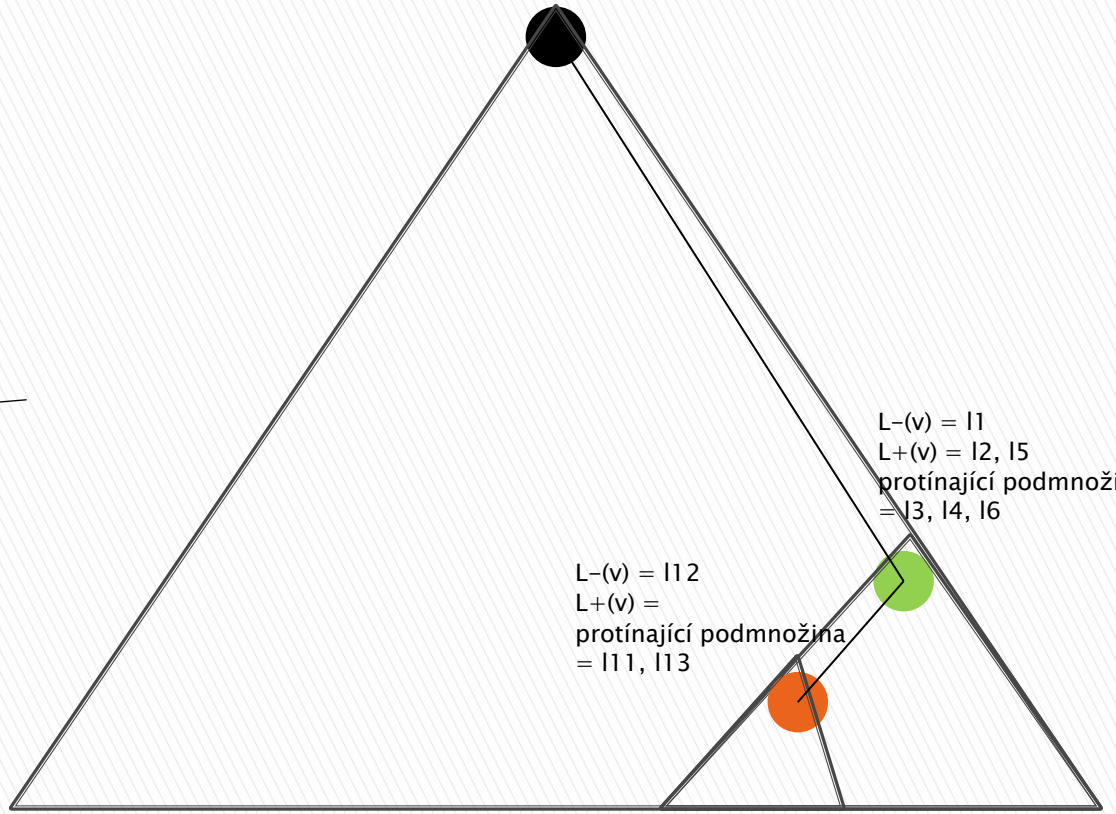
Konstrukce cutting trees





Konstrukce cutting trees





Časová složitost

- Ať Q (n) značí čas dotazu v řezu stromu pro sadu řádků n, pak Q (n) splňuje opakování:

$$Q(n) = \begin{cases} O(1) & if \ n = 1 \\ O(r^2) + Q(n/r) & if \ n > 1 \end{cases}$$

- Tato opakovaní řeší až Q (n) = O (log n) pro všechny konstantní r > 1

$$Q(n) = O(\log n)$$

Časová složitost

- Nechť je dáno ϵ > 0. Můžeme sestrojit (1/r)-cutting pro množinu L velikosti cr^2, kde c je konstanta. Vvtvoříme cutting tree založený na (1/r)-cuttings pro $r = \lceil (2c)^{1/\varepsilon} \rceil$
- Velikost úložiště použité cutting tree, M(n), splňuje

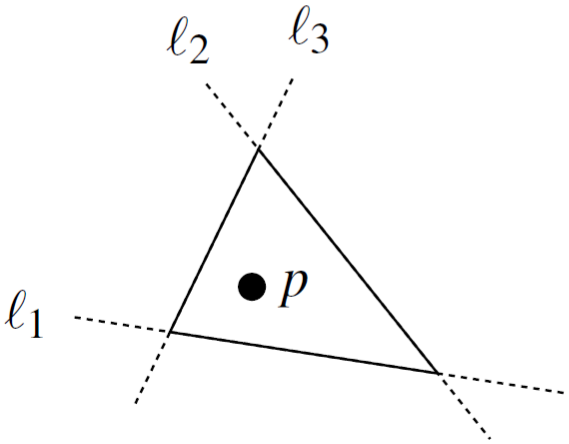
$$M(n) = \begin{cases} O(1) & \text{if } n = 1 \\ cr^2 + \sum_v M(n_v) & \text{if } n > 1 \end{cases}$$

kde sčítáme přes všechny děti v kořene stromu. Počet dětí kořene je cr^2, a nv ≤ n/r pro každé dítě v.

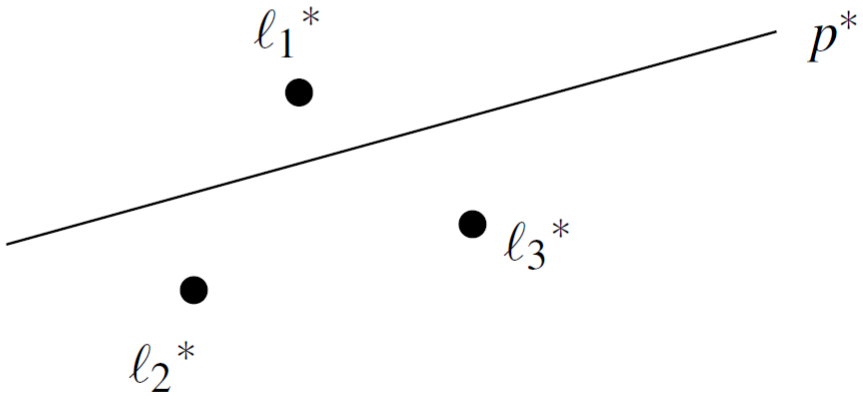
$$M(n) = O(n^{2+\varepsilon})$$

Dual plane

primal plane



dual plane



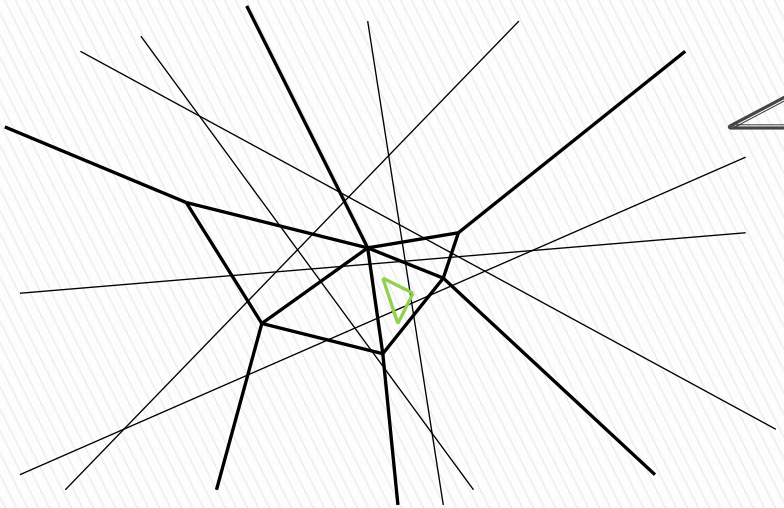
Cutting tree – two level

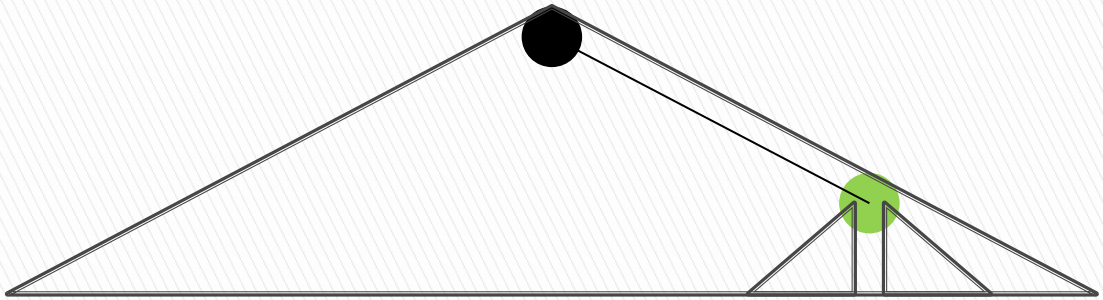
- Vstup: dva body q1, q2 a množina L obsahující n přímek.
- Dotaz: počet přímek z množiny L ležící pod oběma body q1 a q2.

Cutting tree – two level

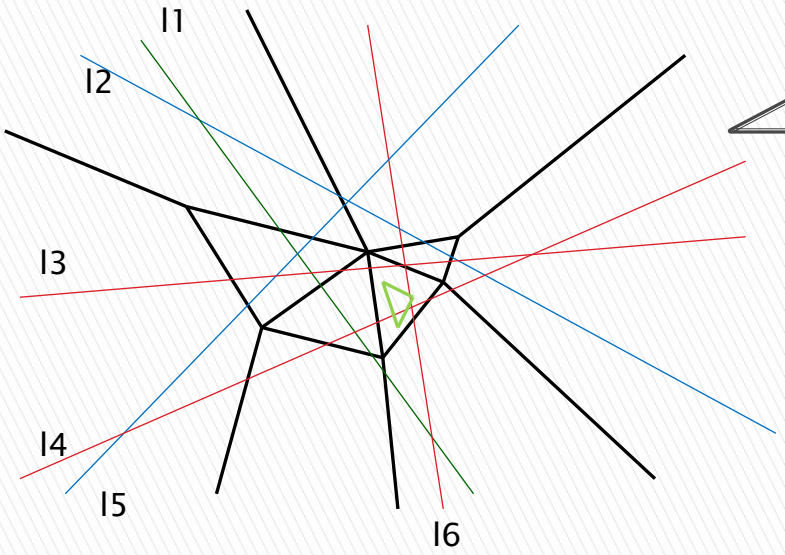
- Množina L je uložena v cutting tree T.
- S každým uzlem v první úrovně stromu T, ukládáme jeho (uzlu v) nižší kanonickou podmnožinu do druhé úrovně cutting tree T

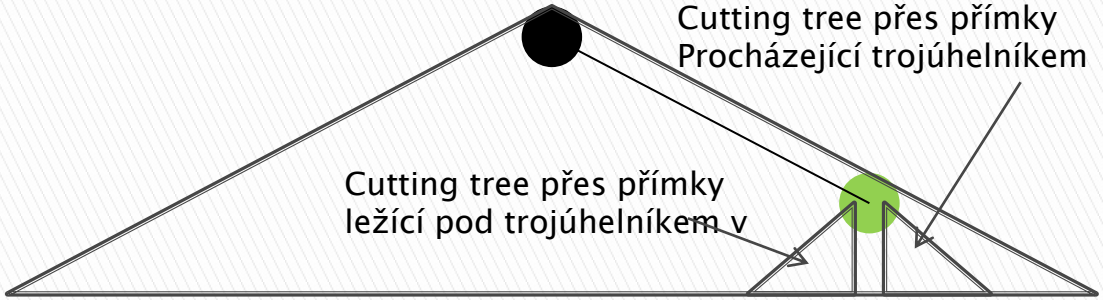
Konstrukce cutting trees–two level





Konstrukce cutting trees–two level





L−(v) = l1

L+(v) = l2, l5

protínající podmnožina = l3, l4, l6

Použití cutting trees–two level

- První úrověn pro hledání přímek pod bodem q1
- Druhá úrověn pro hledání přímek pod bodem q2

Použití cutting trees-two level

Algorithm SELECTBELOWPAIR($q_1, q_2, \mathcal{T}$)
Input. Two query points q_1 and q_2 and a cutting tree or subtree of it.
Output. A set of canonical nodes for all lines in the tree that lie below q_1 and q_2.

1. $\Upsilon \leftarrow \emptyset$
2. **if** $\mathcal{T}$ consists of a single leaf μ
3. **then if** the line stored at μ lies below q_1 and q_2 **then** $\Upsilon \leftarrow \{\mu\}$
4. **else for** each child v of the root of $\mathcal{T}$
5. **do** Check if q_1 lies in $t(v)$.
6. Let v_{q_1} be the child such that $q_1 \in t(v_{q_1})$.
7. $\Upsilon_1 \leftarrow$ SELECTBELOWPOINT$(q_2, \mathcal{T}^{\text{assoc}}_{v_{q_1}})$
8. $\Upsilon_2 \leftarrow$ SELECTBELOWPAIR$(q_1, q_2, \mathcal{T}_{v_{q_1}})$
9. $\Upsilon \leftarrow \Upsilon_1 \cup \Upsilon_2$
10. **return** Υ

Složitost two level tree

- Časová složitost

$$Q(n) = \begin{cases} O(1) & if\ n = 1 \\ O(r^2) + O(\log n) + Q(n/r) & if\ n > 1 \end{cases}$$

$$Q(n) = O(\log^2 n)$$

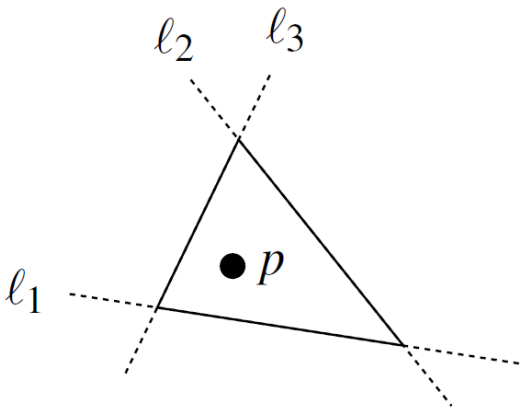
- Paměťová složitost

$$M(n) = O(n^{2+\varepsilon})$$

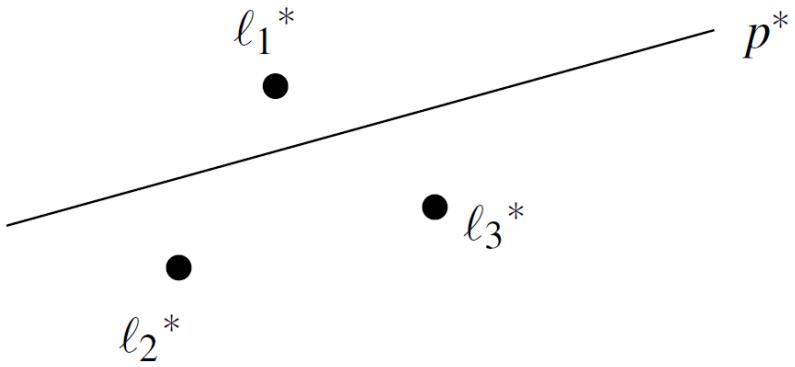
Cutting tree – three level

- Vstup: tři body q1, q2, q3 a množina L obsahující n přímek v rovině.
- Dotaz: počet přímek z množiny L ležící pod nebo nad body q1, q2, q3, tak aby splňovala kritéria.

primal plane



dual plane



Literatura

- [1] Computational Geometry – Algorithms and Applications, 3rd Ed
- [2] Half-Space Range Searching – Piyush Kumar
- [3] CS268: Geometric Algorithms Handout #13, Design and Analysis Original Handout #13, Stanford University Tuesday, 24 May 1994, Original Lecture #13: Tuesday, May 24, 1994, Topics: Range Searching, Efficient Partition Trees for Range Searching
- [4] Geometric range searching – Jiří Matoušek
- [5] Windowing queries, Cutting Tree, R. Inkulu, http://www.iitg.ac.in/rinkulu/
- [6] Cutting trees and the cutting lemma – Amir Mograbi

Dotazy?

Děkuji za Pozornost

Martin Chaloupka
chaloma3@fel.cvut.cz

OI-OPPA. European Social Fund
Prague & EU: We invest in your future.