

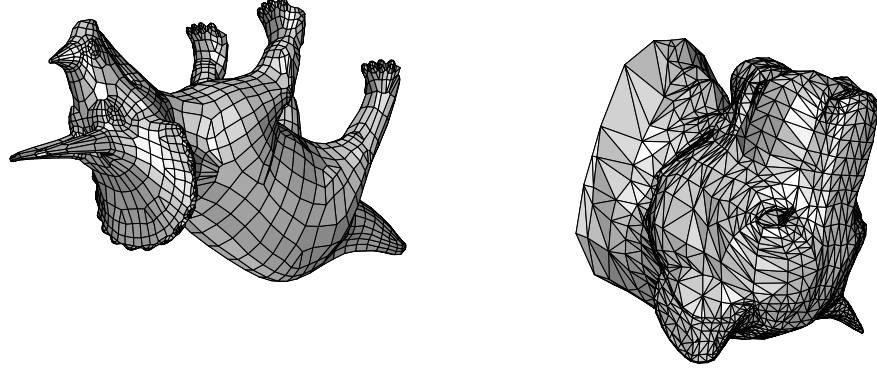
Triangulation canonique d'un graphe
planaire 3-connexe, avec une application de
codage

Luca Castelli Alvardi

23 septembre 2004

Compression de maillages polygonaux

- compression sans perte de la connectivité du maillage;
- graphes planaires 3-connexes: maillages polygonaux équivalent topologiquement à une sphère (genre 0, sans bords, une seule composante connexe)



Maillages polygonaux

Entropie de Tutte (1963)

- enumeration des graphes planaires 3-connexes à m arêtes:

$$\Phi_m \approx \frac{9}{2} \sqrt{\frac{\pi}{6}} m^{-\frac{21}{2}} 4^m \quad (1)$$

- valeur de l'entropie exprimée en bpe (bits/arête):

$$e = \frac{1}{m} \log_2 \Phi_m \text{ bpe}$$

borne supérieure théorique $e = \log_2 4 = 2 \text{ bpe}$ (2)

$$\text{Entropie de } n \text{ symboles: } \mathcal{E} := \sum_{i=1}^n d_i \log_2 \frac{1}{d_i} \quad (3)$$

Compression de graphes 3-connexes: travaux récents

Algorithmes avec borne supérieure garantie

- Keeler et Westbrook ('95): 3 *bpe*

- He et. al ('99): 2.83 *bpe*

- Chuang et. al ('98): 2.37 *bpe*

- Fusy, Poulalhon et Schaeffer (2004): toujours 2 *bpe* (optimal)

Méthodes euristiques sans borne supérieure garantie

(algorithmes basés *valence/degre*, presque optimaux)

- triangulations: Touma-Gotsman ('98), Alliez-Desbrun ('01)

- maillages polygonaux: Khodakovsky et al. ('02), Isenburg ('02)

En pratique taux de compression $> 2 \text{ bpe}$

Méthodes à la "TG" (Touma-Gotsman '98)

basées *degré/valence*

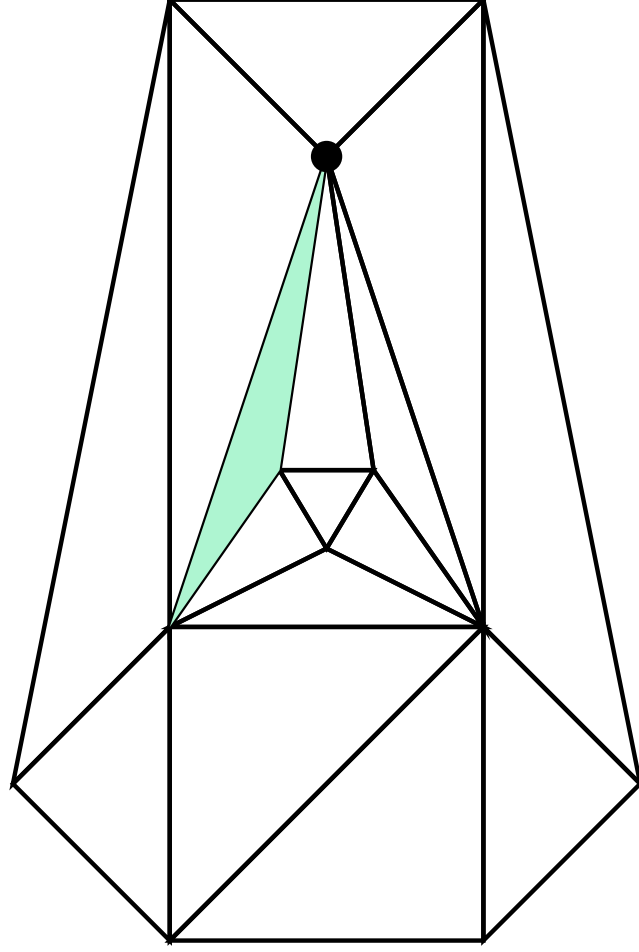
- définition d'une liste active: sommets d'un cycle séparateur;
- algorithme par expansion de la liste active;
- conquête des arêtes autour du sommet *Focus* en sens trigonométrique;

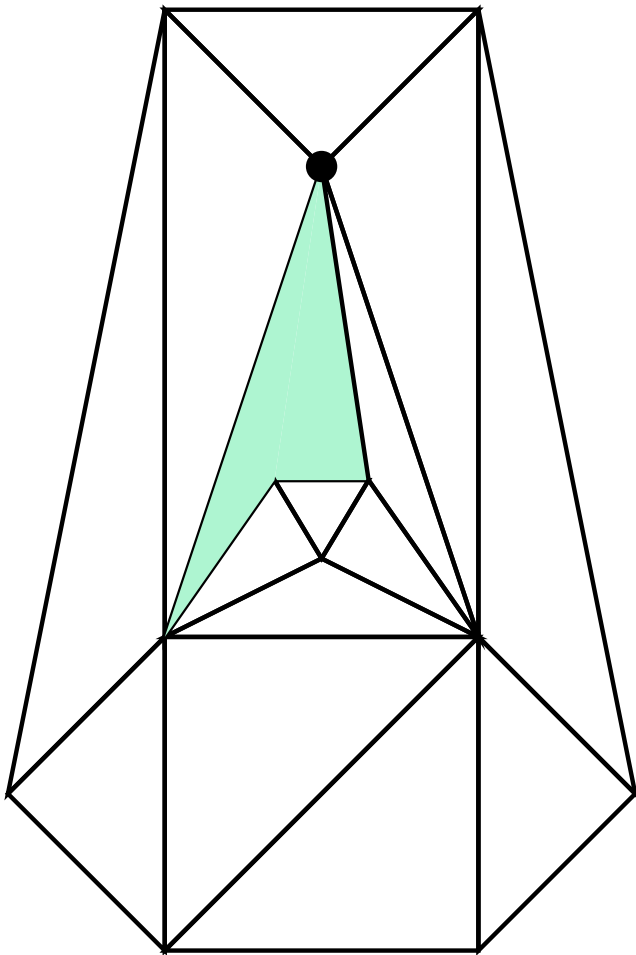
- codage à l'aide des degrés des faces;
- auto-intersection de la liste active: code *split*.

Remarques

- bon taux de compression pour maillages réguliers;
- aucune borne théorique intéressante;
- presque optimale: taux de compression de 3.24, si le nombre de *split* est négligeable.

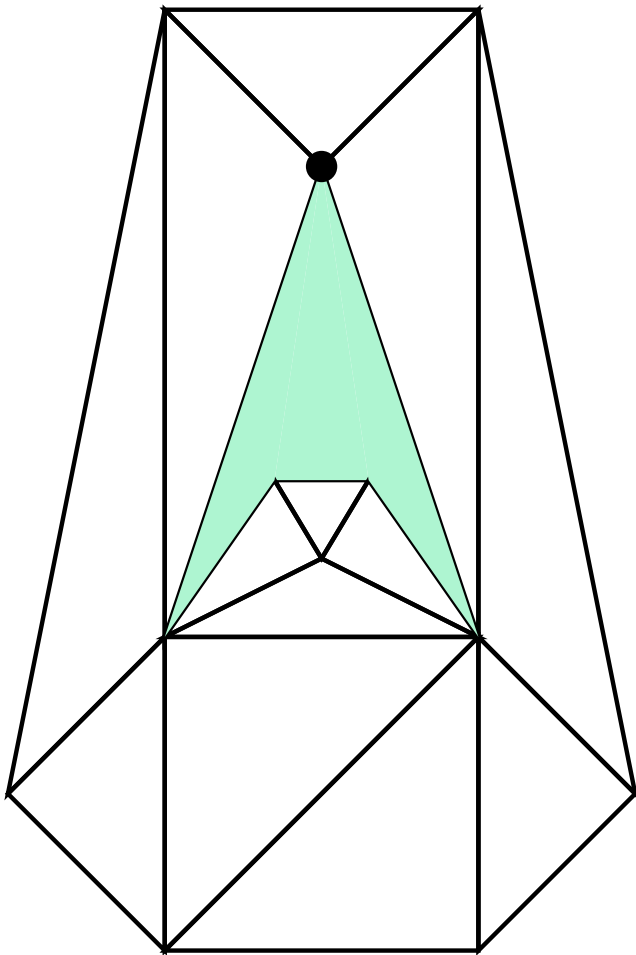
Algorithme TG (degré des sommets)



$\text{add}(4)$ 

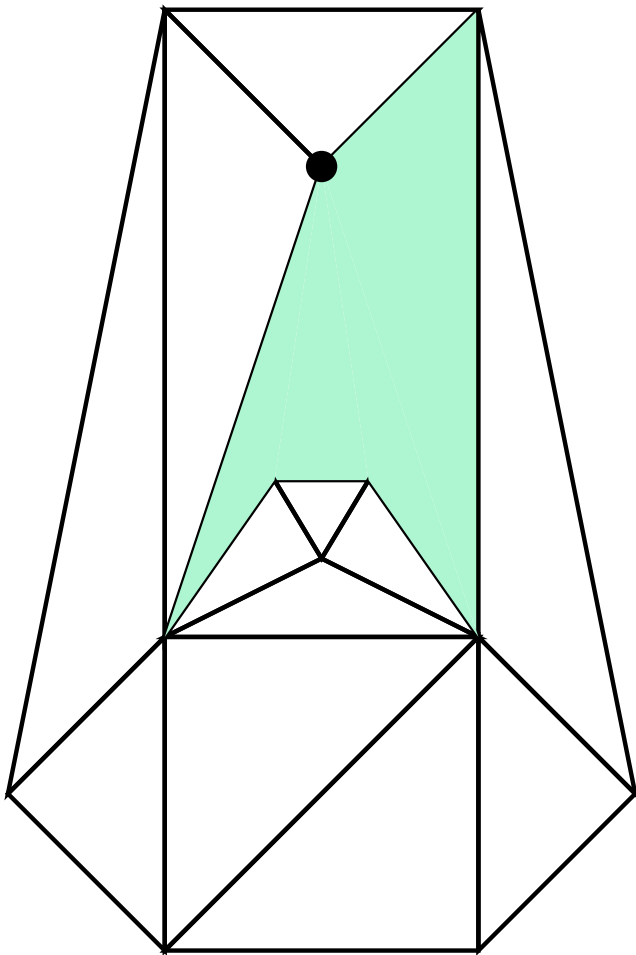
Algorithme TG (degré des sommets)

(8)ppø



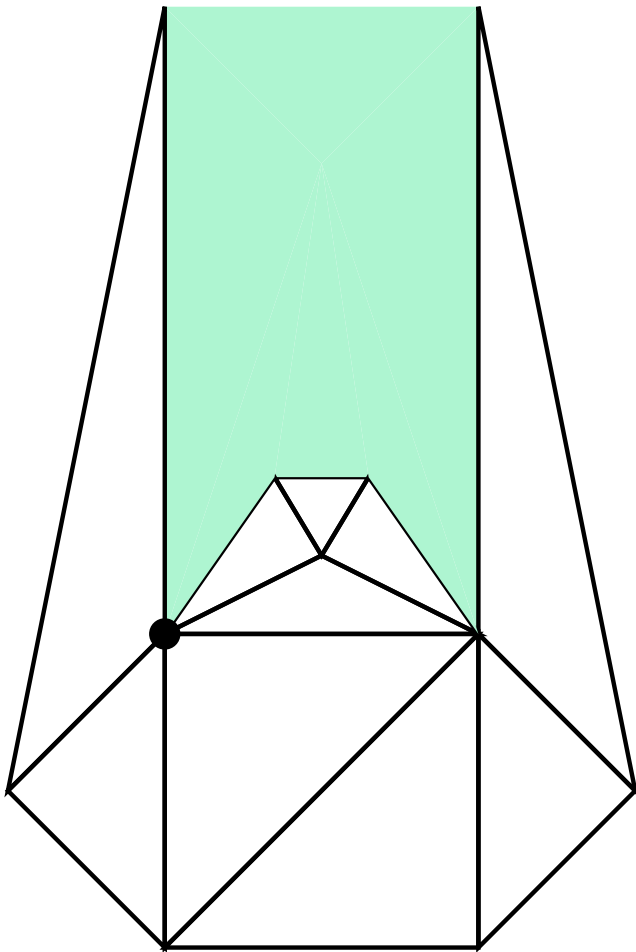
Algorithme TG (degré des sommets)

(c)ppæ



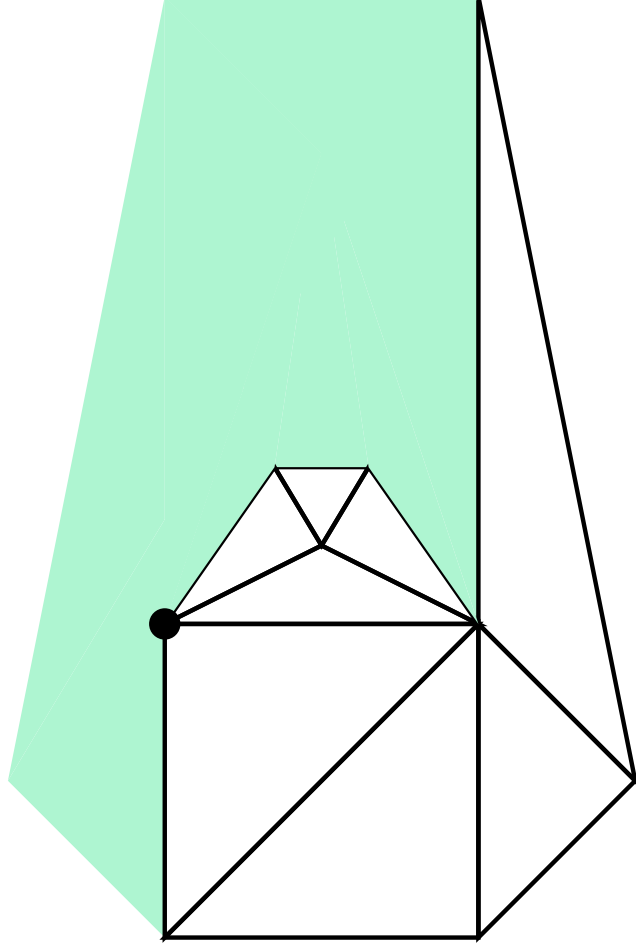
Algorithme TG (degré des sommets)

add(4), add(5)



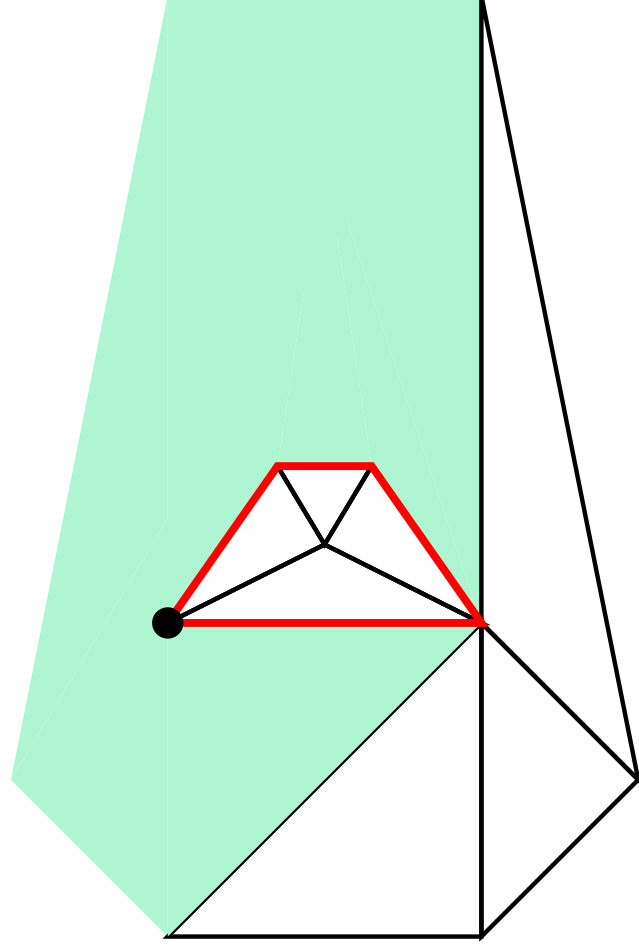
Algorithme TG (degré des sommets)

Algorithme TG (degré des sommets)



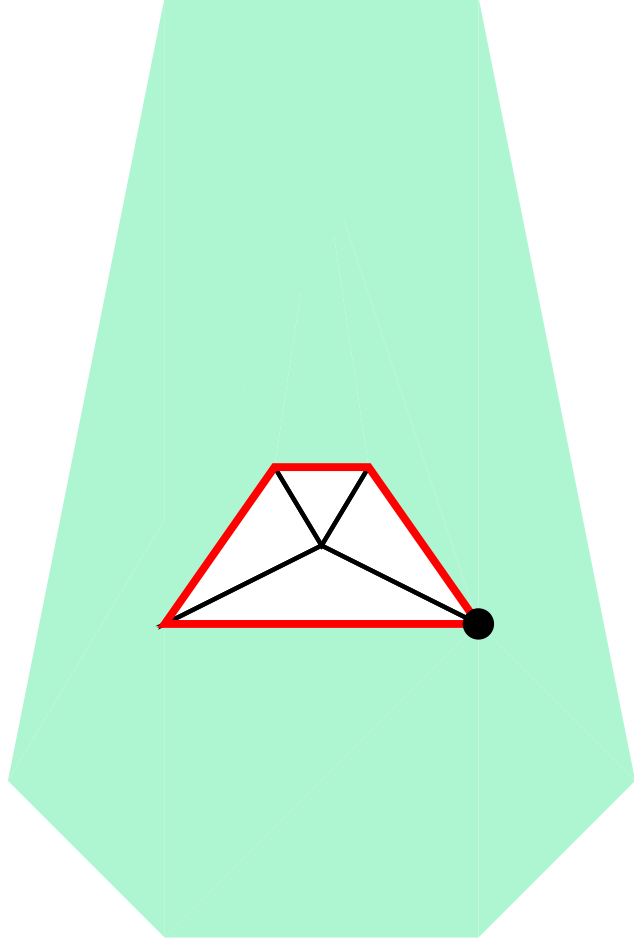
add(4), add(5)

Algorithme TG (degré des sommets)



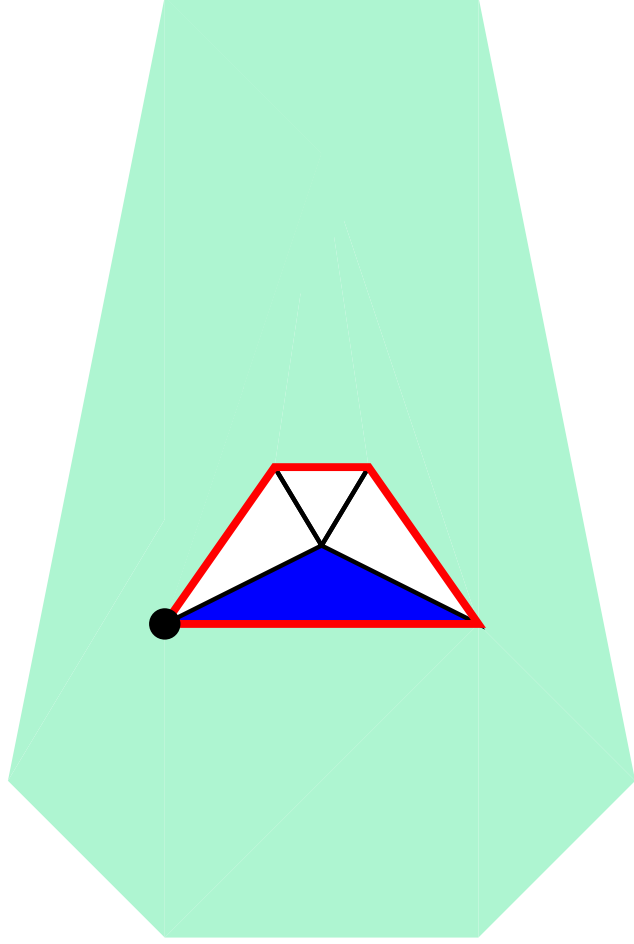
split 5

Algorithme TG (degré des sommets)



$\text{add}(4), \text{add}(4)$

Algorithme TG (degré des sommets)



add(4)

Codage des graphes planaires 3-connexes

Algorithme en deux phases, basé sur les propriétés des ordres canoniques:

- calculer une triangulation canonique T_G of G ;
- on utilise seulement la connectivité de T_G et la suite des degrés des faces pour coder le graph original G ;
- coder T_G avec un codeur pour maillages triangulaires;

Idées principales de l'algorithme:

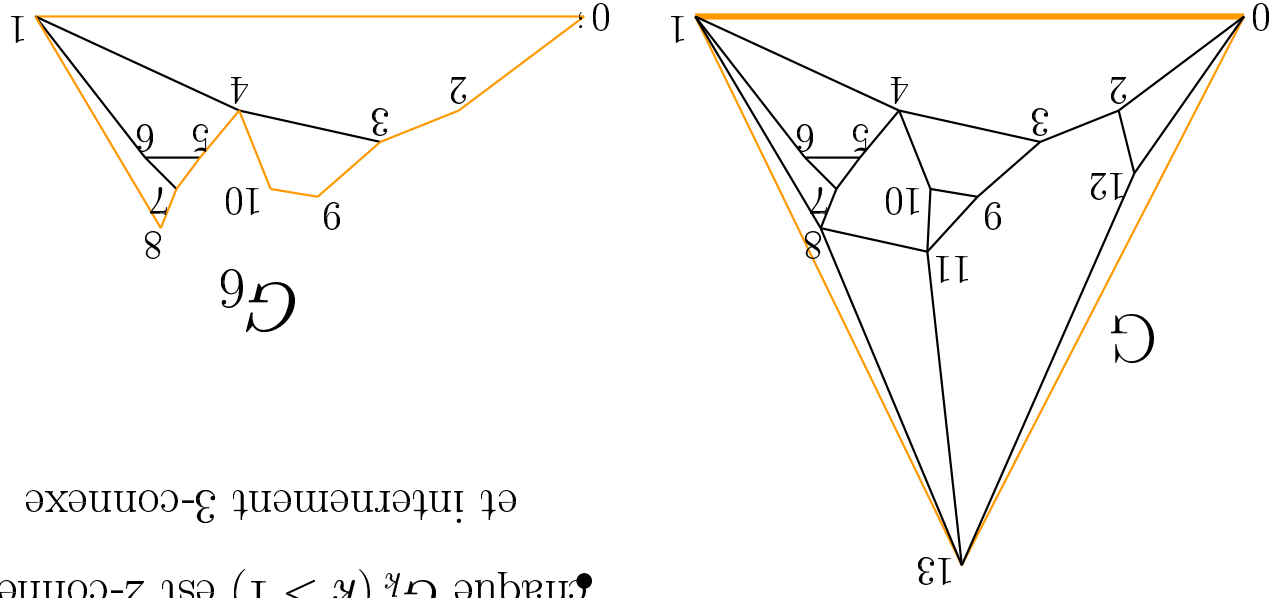
- on fait un parcours du graphe par conquête des sommets: sans *splits*;
- presque-optimalité des méthodes basées *valence/degre*;
- un codage optimale de la triangulation nécessite de $3.24bpv$;
- borne supérieure garantie: $2.62 bpv$;
- algorithme adaptatif: bons taux de compression pour les maillages réguliers.

Ordre canonique pour graphes planaires 3-connexes

Définition (Kant'96). Soit $\pi = \{V_1, \dots, V_k\}$ une partition ordonnée des sommets V . Notons G_k le sous-graphe de G induit par $V_1 \cup \dots \cup V_k$ et C_k le bord de la face externe G_k (arêtes oranges). On dit que π est un *ordre canonique* de G si :

- $V_1 = \{v_0, v_1\}$, $V_k = \{v_{n-1}\}$ et chaque C_k est un cycle contenant (v_0, v_1) ;

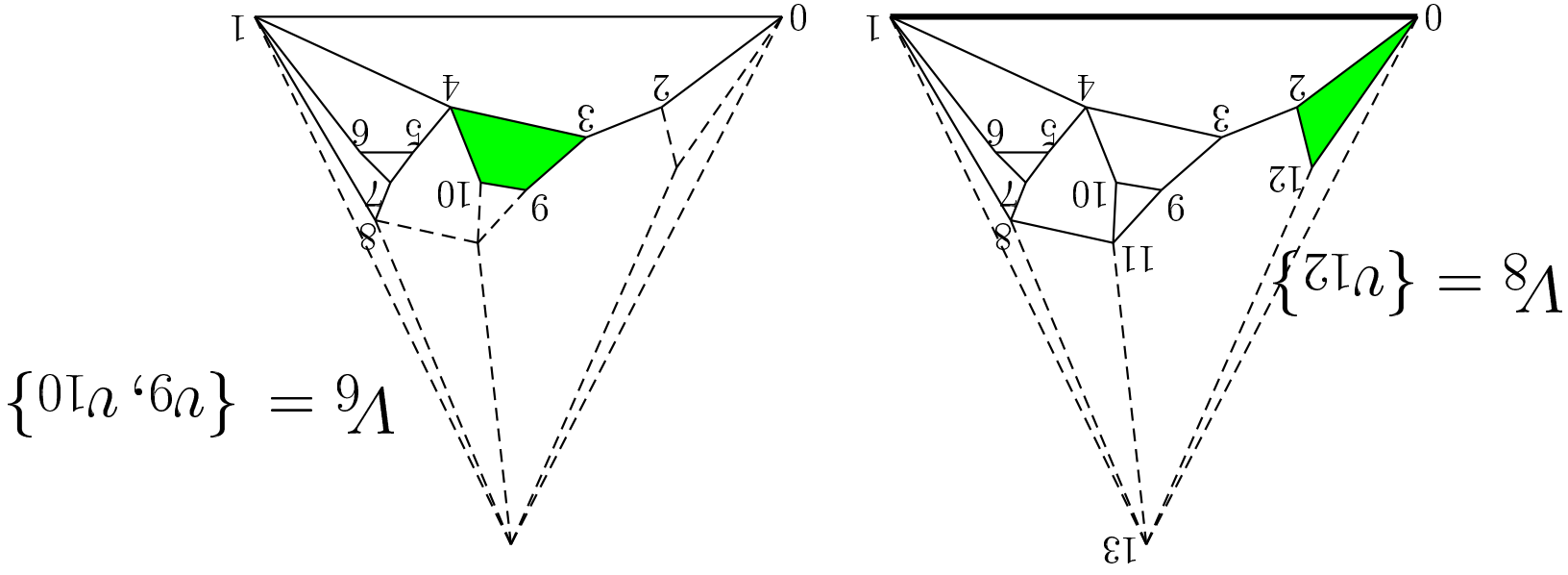
- chaque G_k ($k > 1$) est 2-connexe et internement 3-connexe



Ordre canonique pour graphes planaires 3-connexes

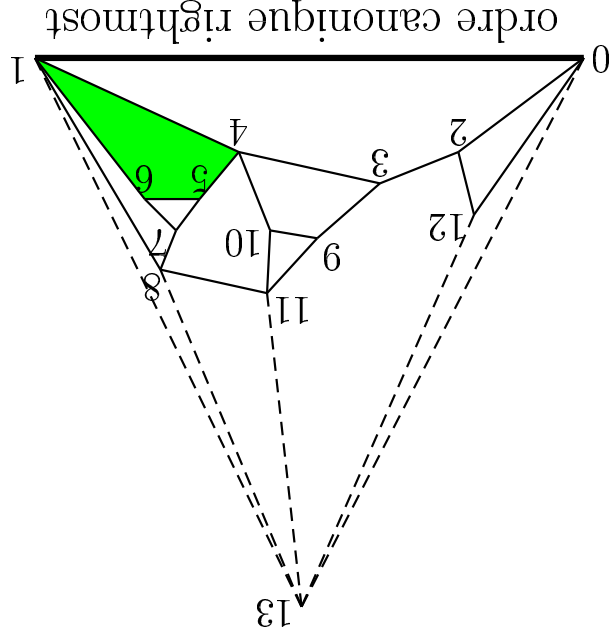
Pour tout $k \in \{2, \dots, K-1\}$ l'une des deux conditions suivantes doit être satisfaite :

1. $V_k = \{z\}$ t.q. $z \in C_k$ et z a au moins un voisin dans $G - G_k$
2. $V_k = \{z_1, \dots, z_l\}$, chaque z_i a au moins un voisin dans $G - G_k$ et z_1, z_l a exactement un voisin sur C_{k-1} .

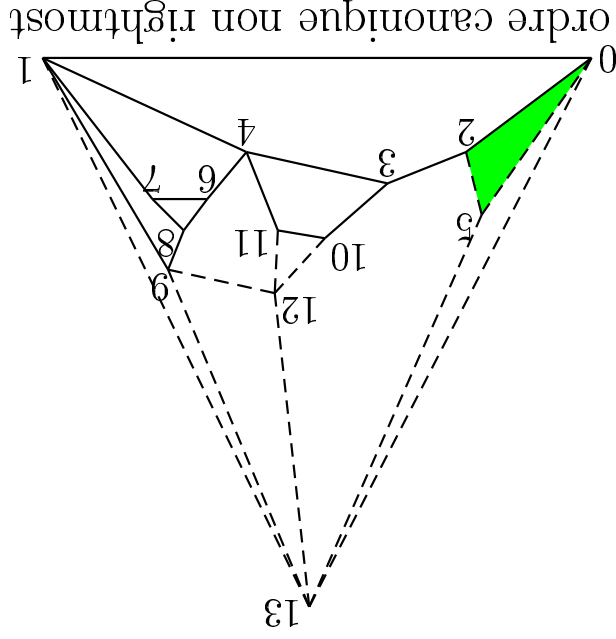


Ordre canonique Rightmost

Définition. (intuitivement) Un ordre canonique est *rightmost* si chaque fois qu'il existe plusieurs chaînes V_k que l'on peut ajouter à G_{k-1} , alors on ajoute celle plus à droite.



$V_1 = \{0, 1\}, V_2 = \{2, 3, 4\}, V_3 = \{5, 6\}$



$V_1 = \{0, 1\}, V_2 = \{2, 3, 4\}, V_3 = \{5\}$

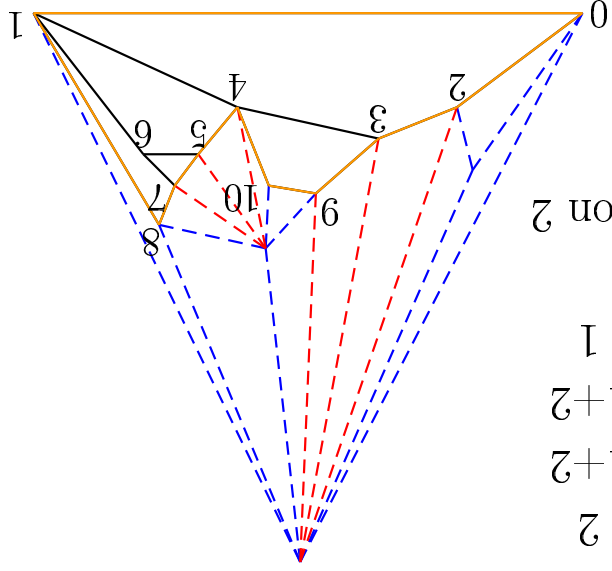
ordre canonique non rightmost

Notation: sommet libre

1. v est incident à au moins une arête bleu;

2. sa suppression (avec celle de ses arêtes dans G_k) ne change pas la 2-connexité du sous-graphe G_k .

v_2 non libre: condition 2
 v_3 non libre: condition 1+2
 v_4 non libre: condition 1+2
 v_7 non libre: condition 1
 v_8 libre:
 v_9, v_{10} non libres: condition 2



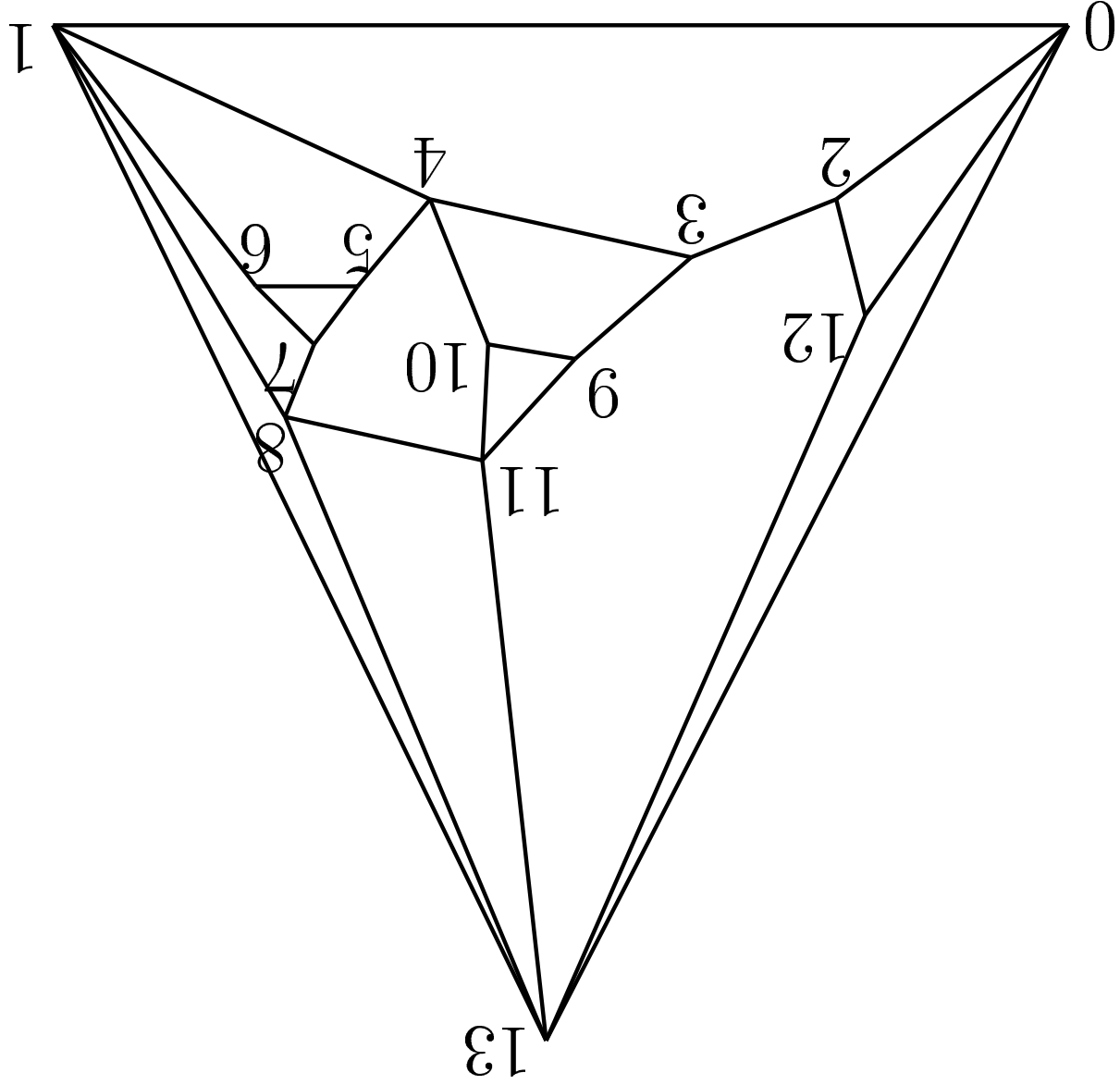
Codage/décodage de graphes planaires 3-connexes

Phases principales de l'algorithme

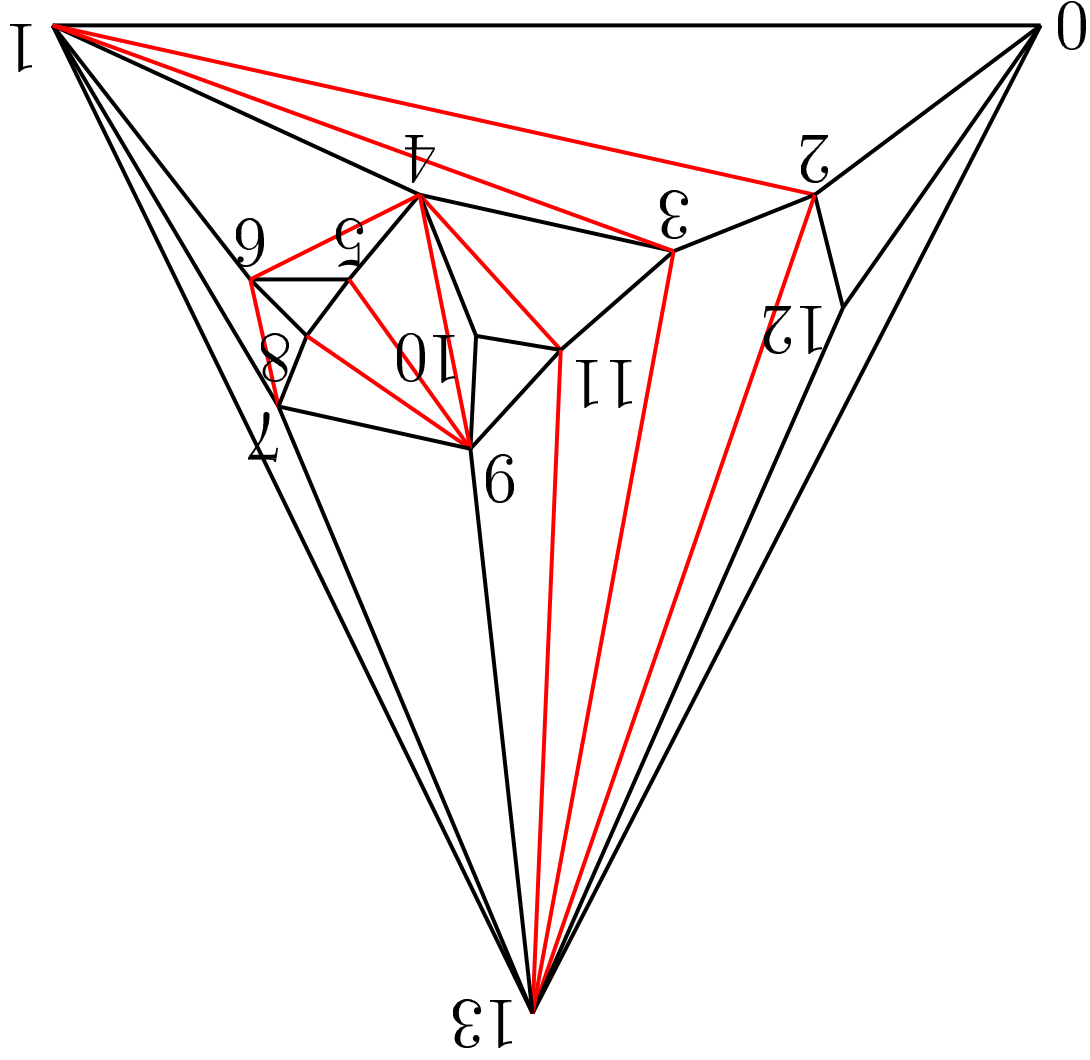
- on calcule une triangulation T of G ;
 - compression de T à l'aide d'un codeur (optimal) pour triangulations;
 - on fait un parcours de T selon une version modifiée de l'ordre canonique right-most;
 - On écrit (ou on lit) les degrés des faces de G visitées pendant le parcours (visitées pour la première fois);
 - la suite de degrés des faces est codée à l'aide d'un codeur arithmétique.
- Caractéristique de notre approche:

- Il s'agit d'une conquête par "retraction" du graph G ;
- à chaque étape on supprime un sommet libre (le plus à gauche) avec ses arêtes et faces incidentes (visitées en sens direct);

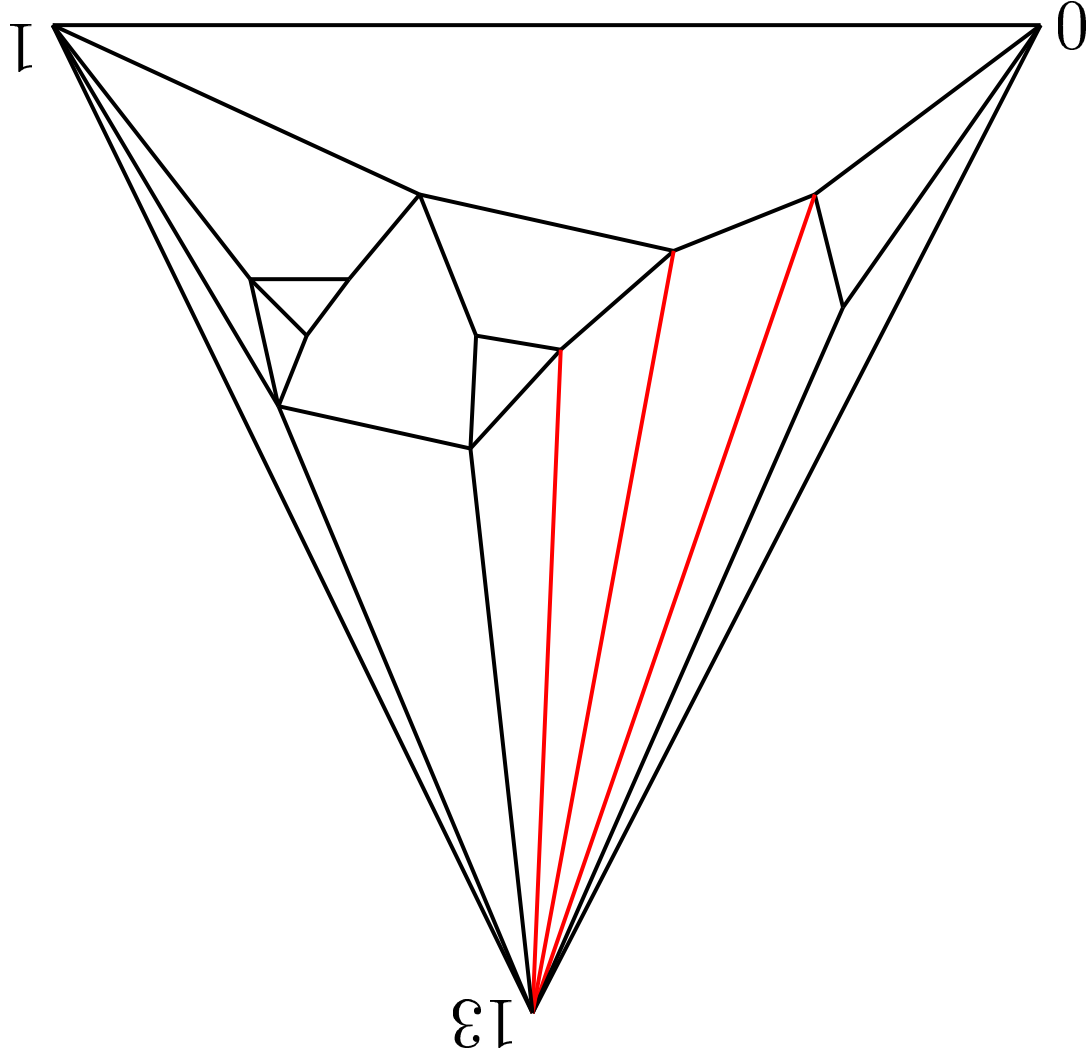
Ordre canonique right-most de G



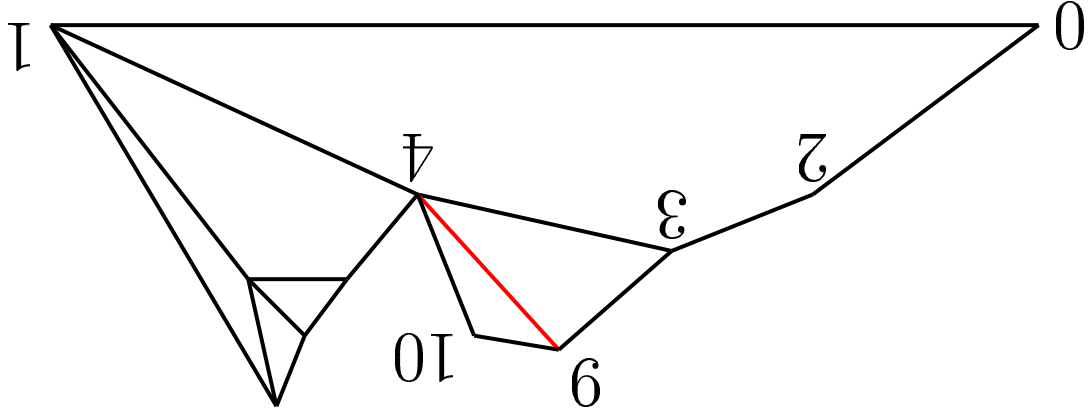
Triangulation canonique G (avec son ordre canonique right-most)



Triangulation des faces de G
cas 1: autour d'un seul sommet (v_{13})



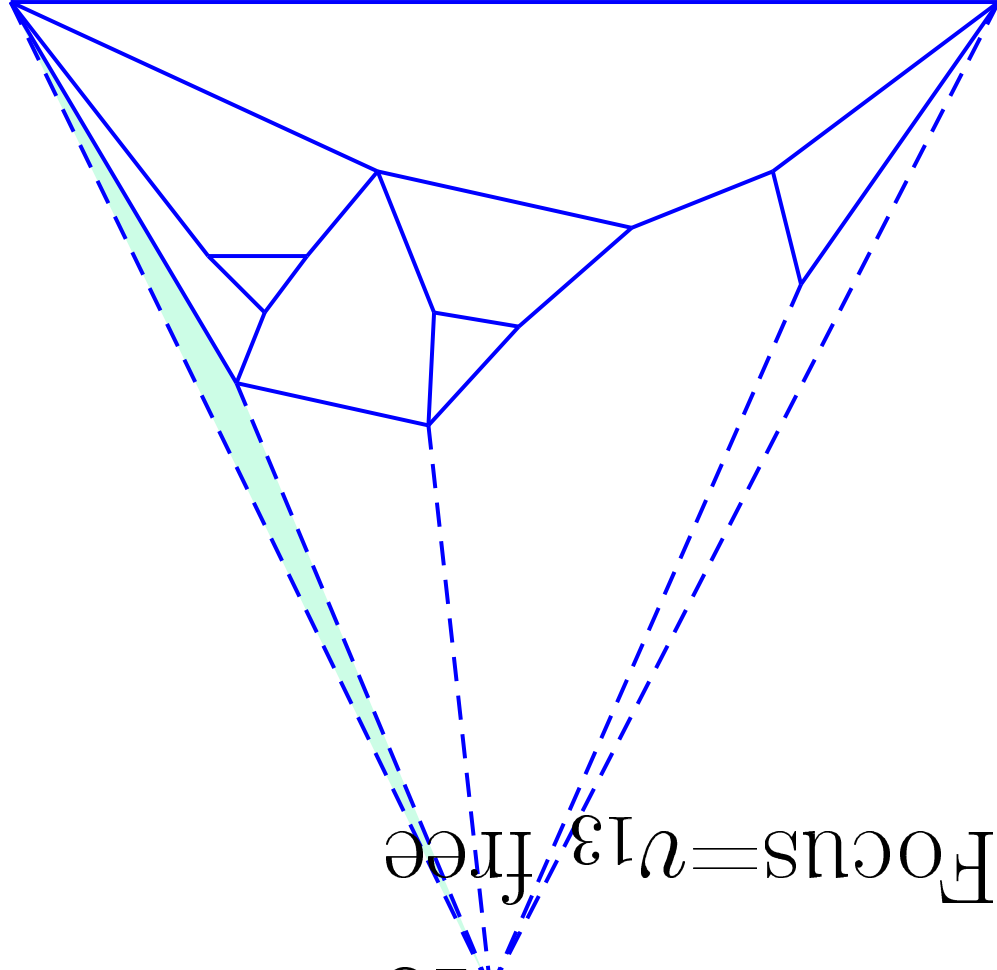
cas 2: triangulation d'une face (v_{10}, v_9, v_3, v_4)
on ajoute l'étoile d'arêtes sortantes de v_9



Calcul de la triangulation

13

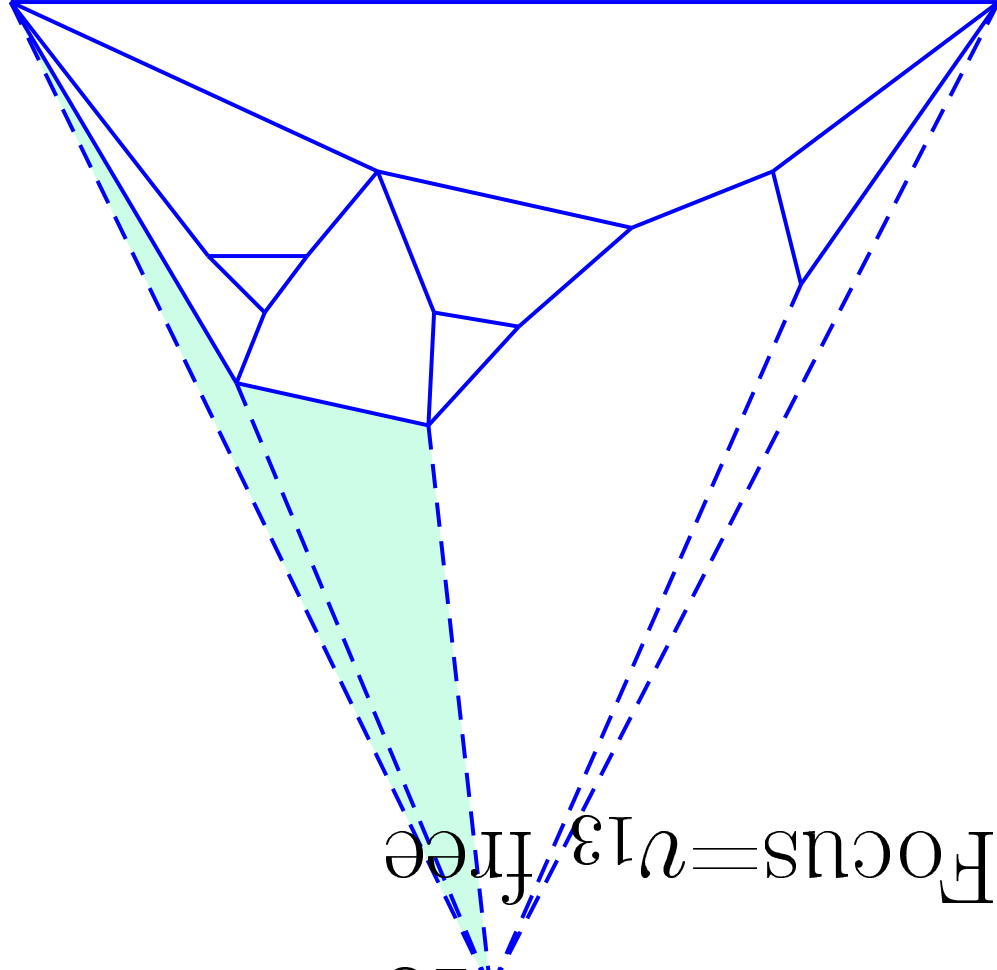
Focus = v_{13} free



Calcul de la triangulation

13

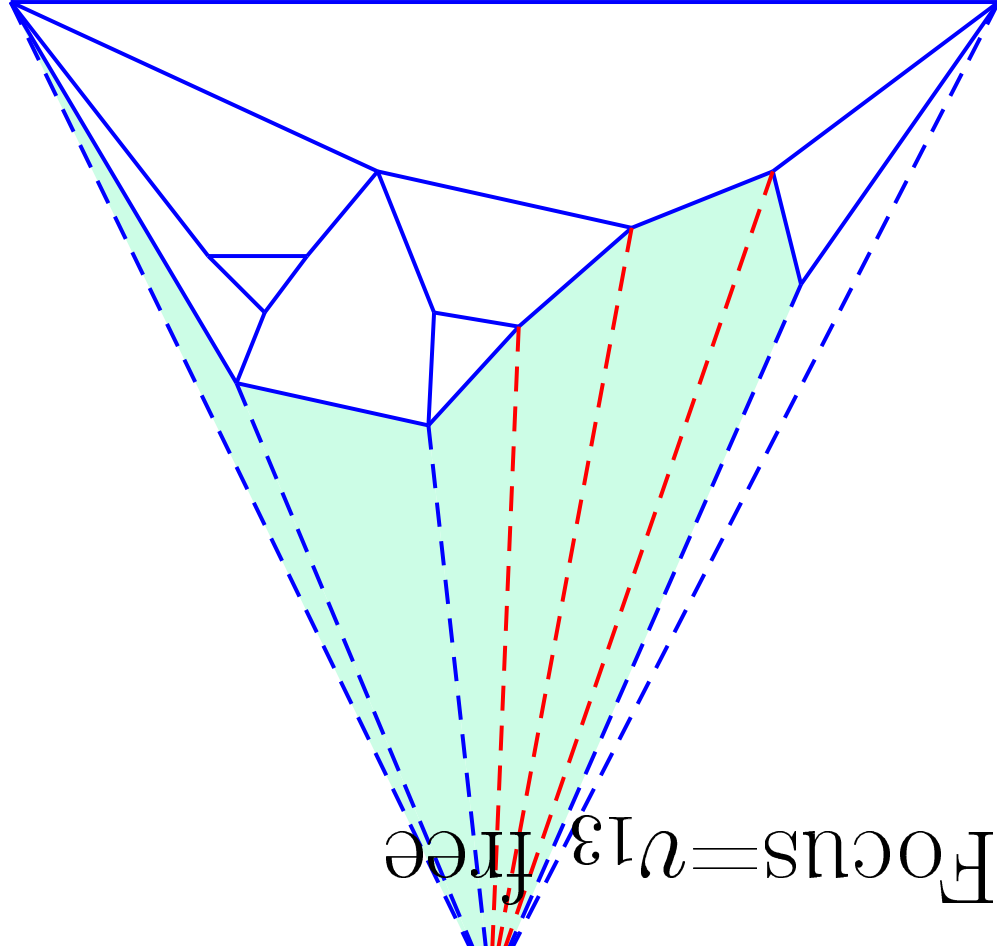
Focus = v_{13} free



Calcul de la triangulation

13 •

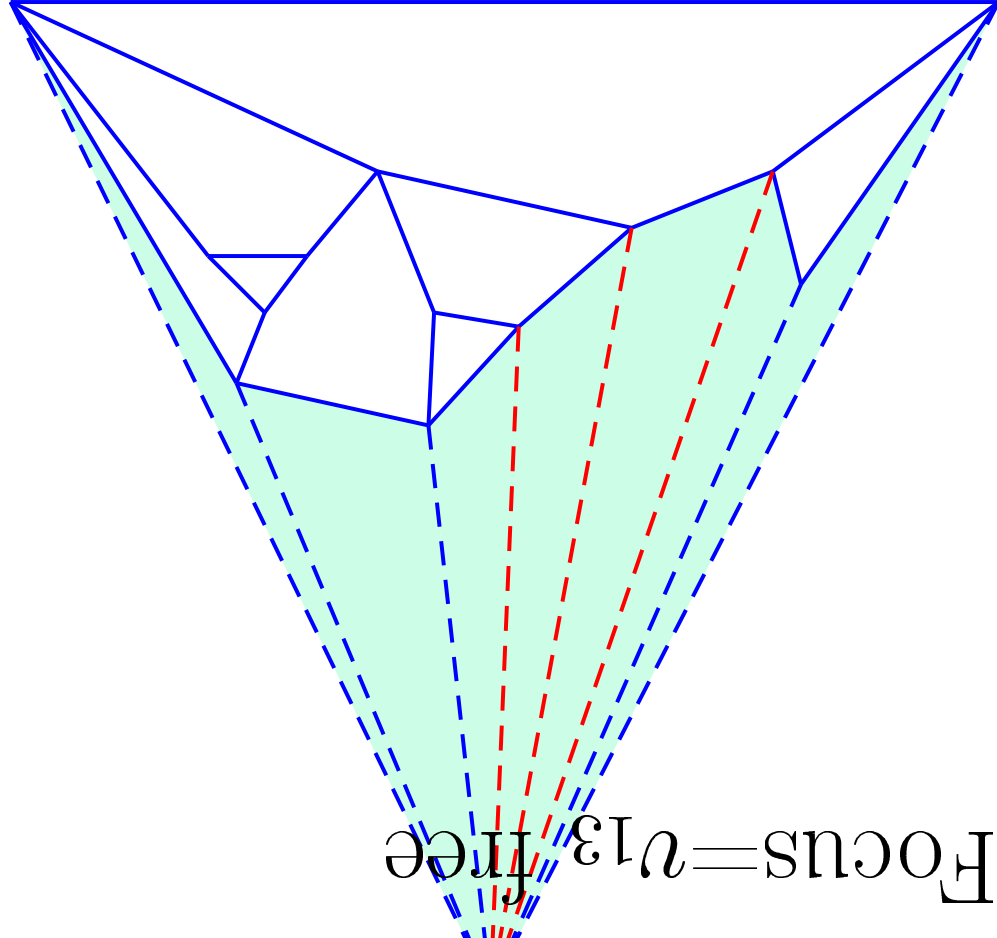
Focus = v_{13} free



Calcul de la triangulation

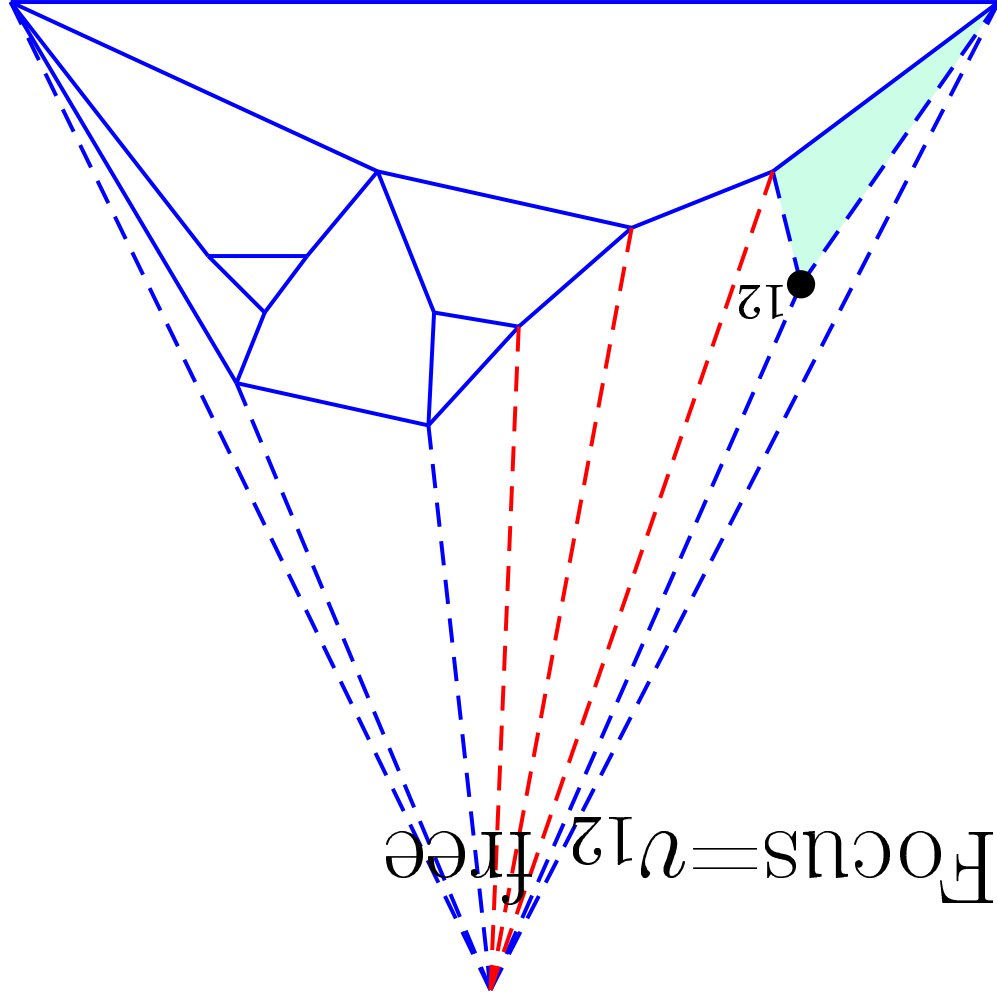
13 •

Focus = v_{13} free



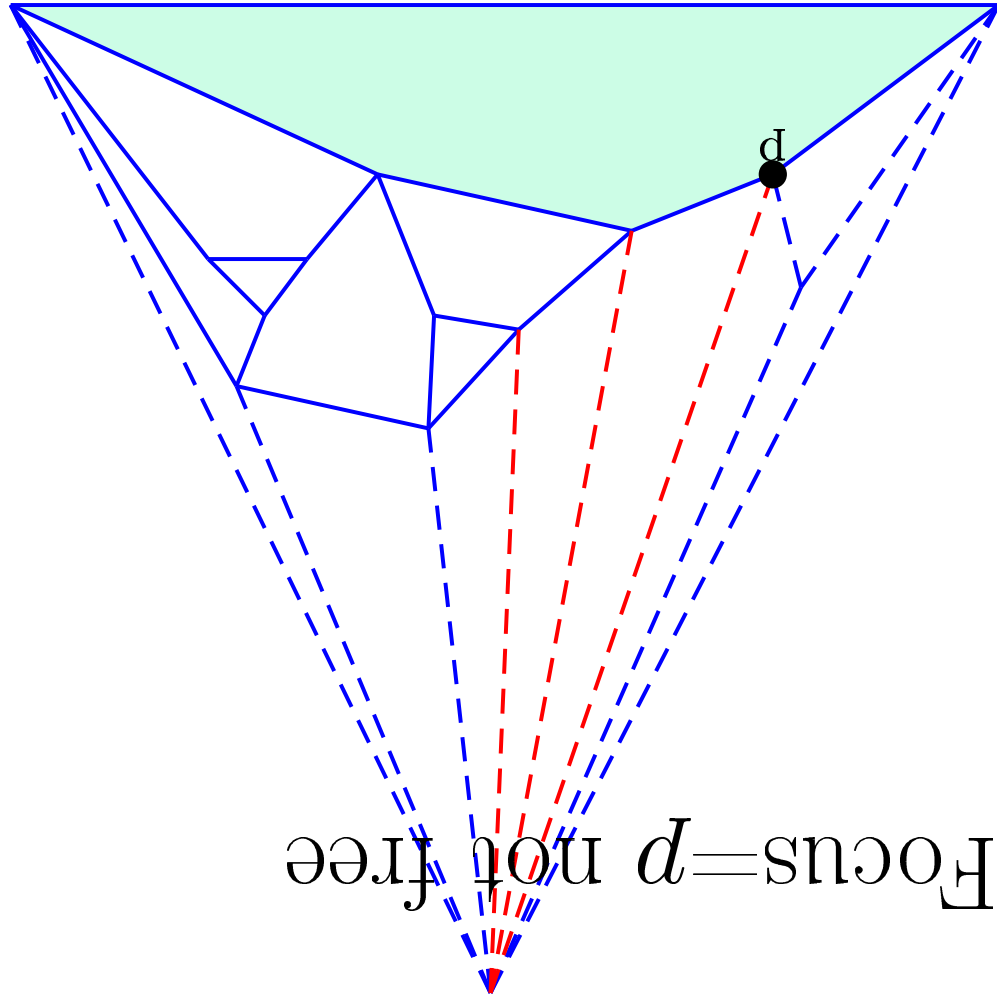
Calcul de la triangulation

Focus = v_{12}



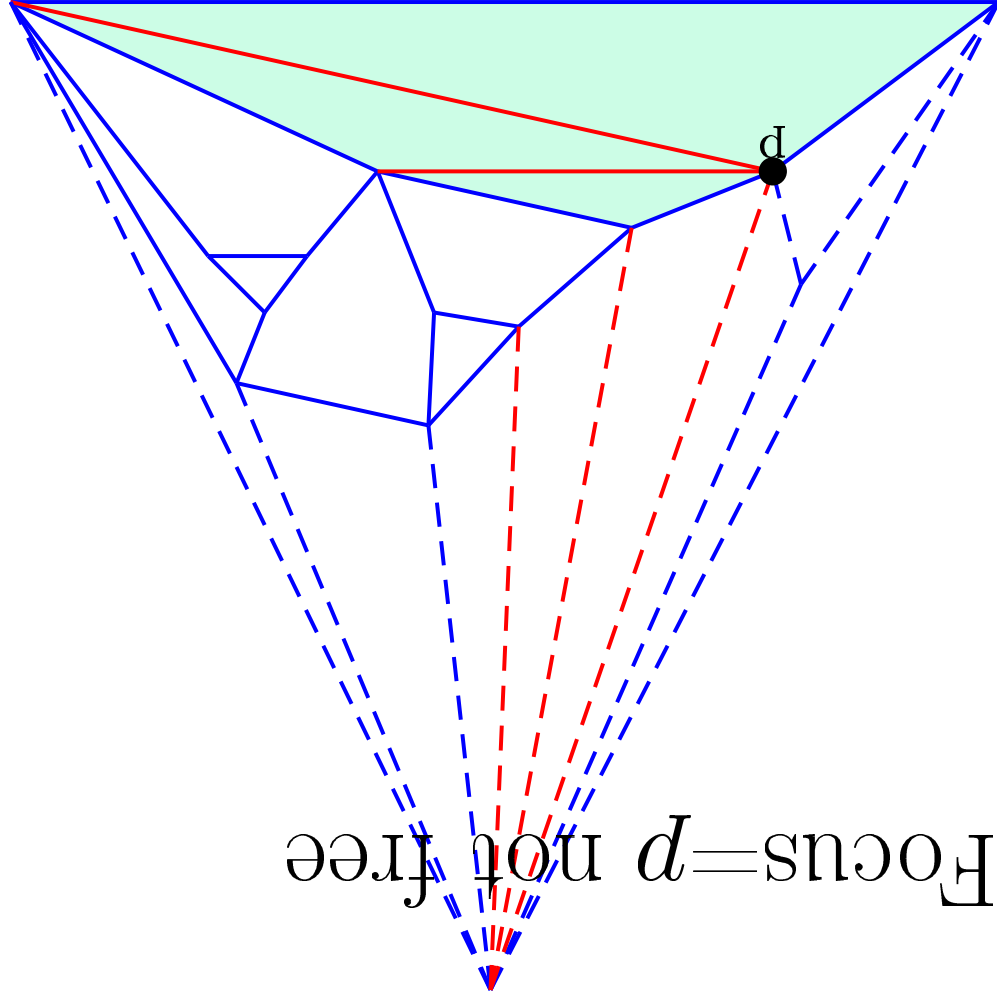
Calcul de la triangulation

Focus = p not free



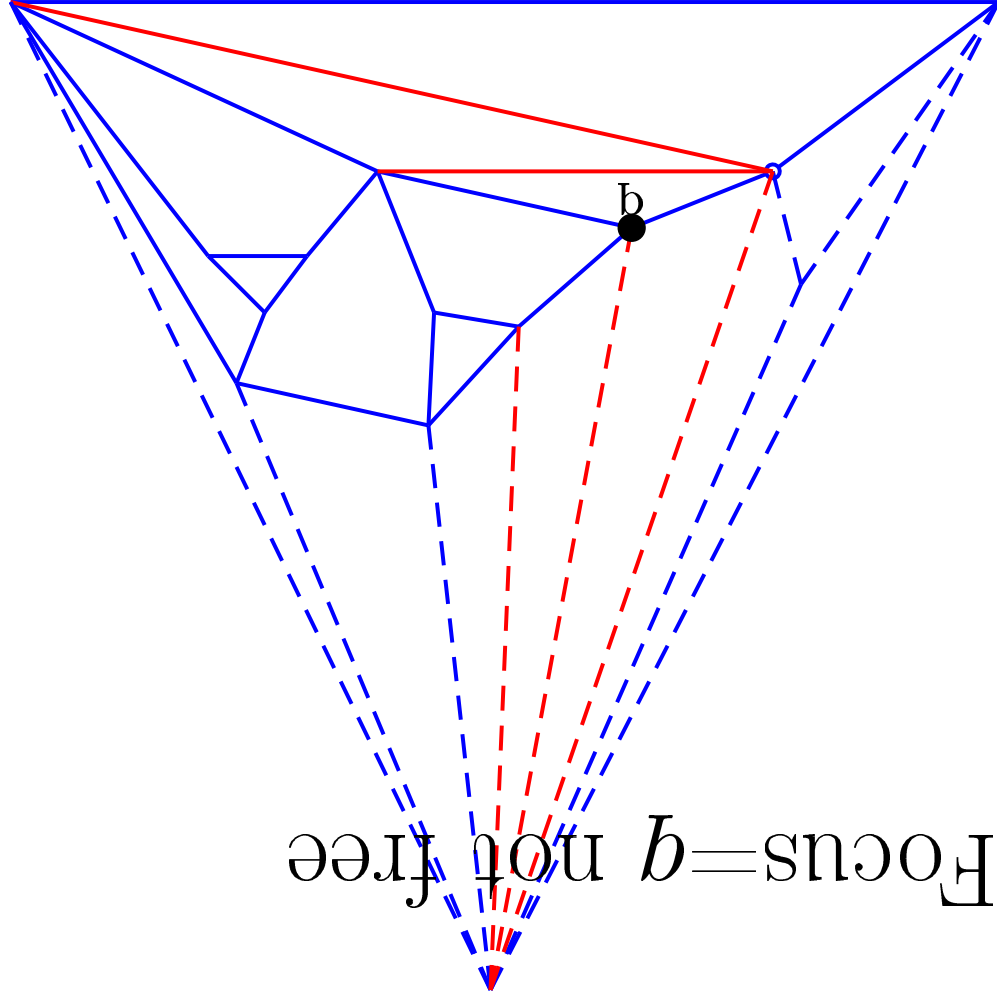
Calcul de la triangulation

Focus = p not free



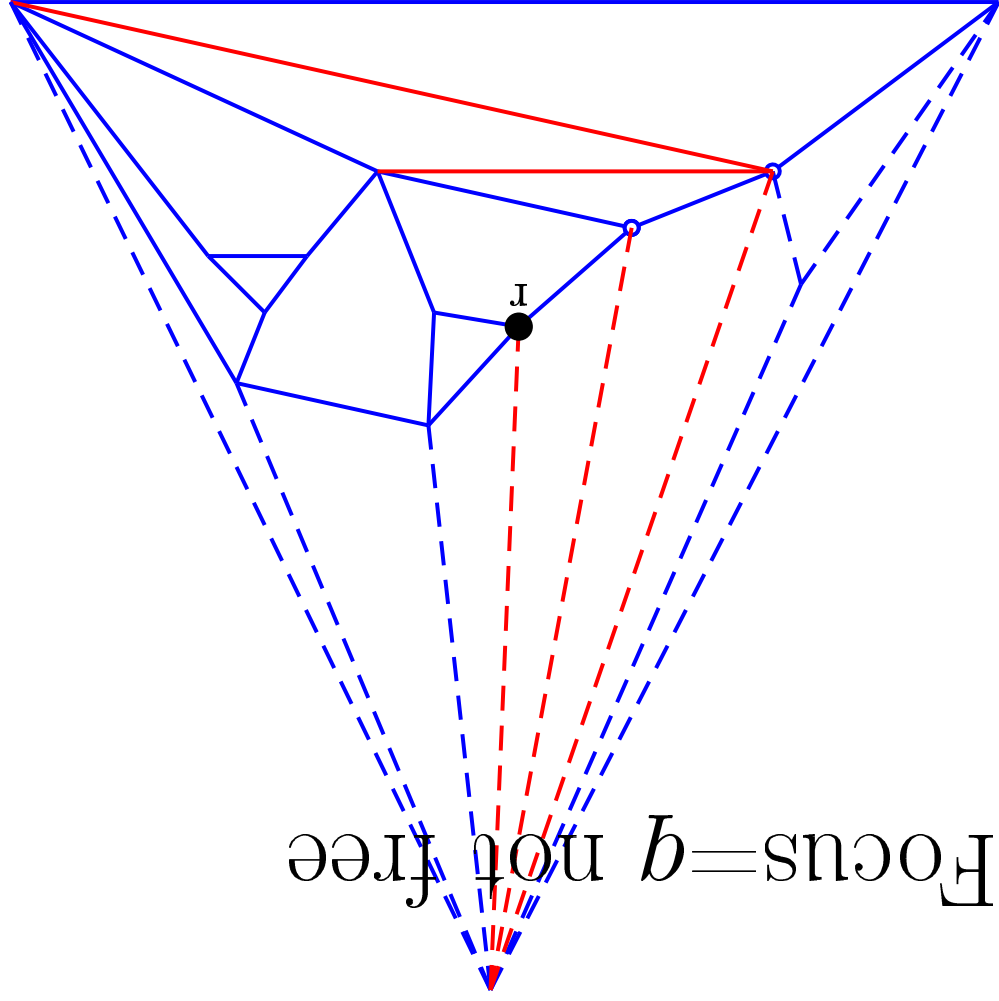
Calcul de la triangulation

Focus = q not free



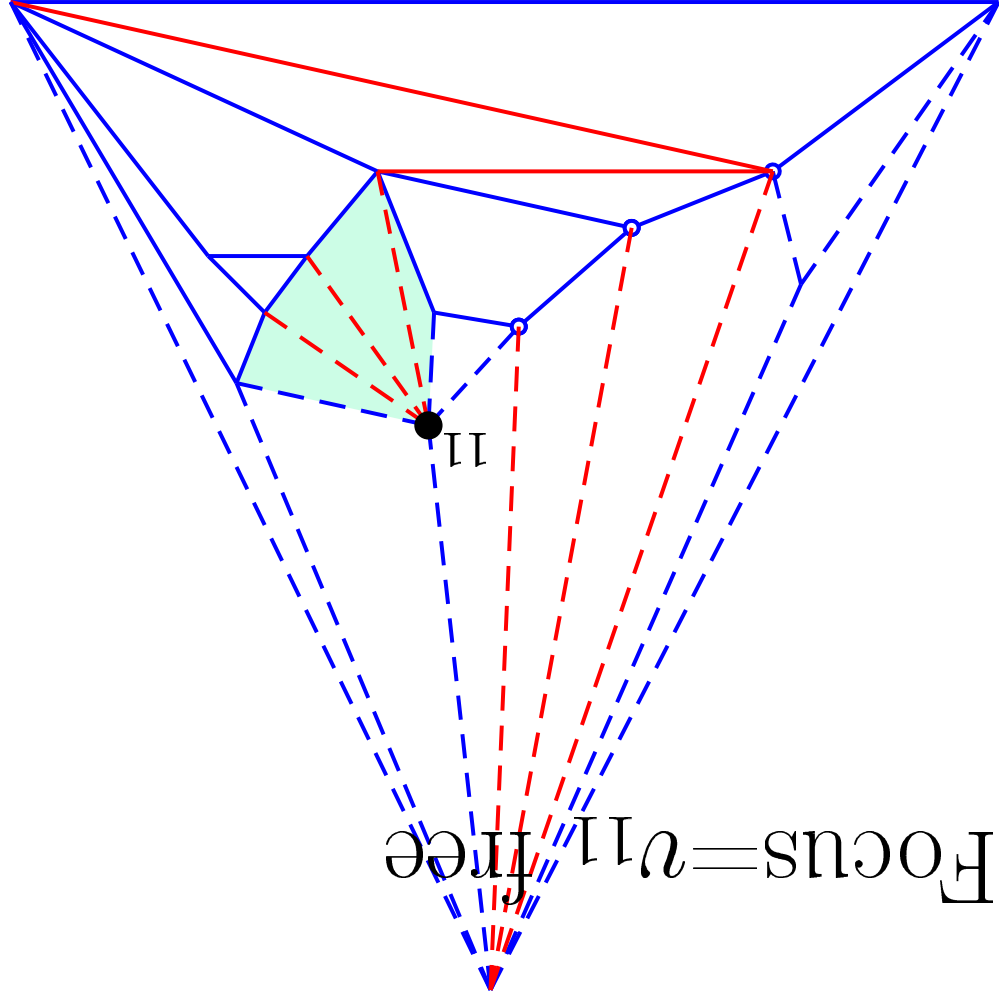
Calcul de la triangulation

Focus = q not free



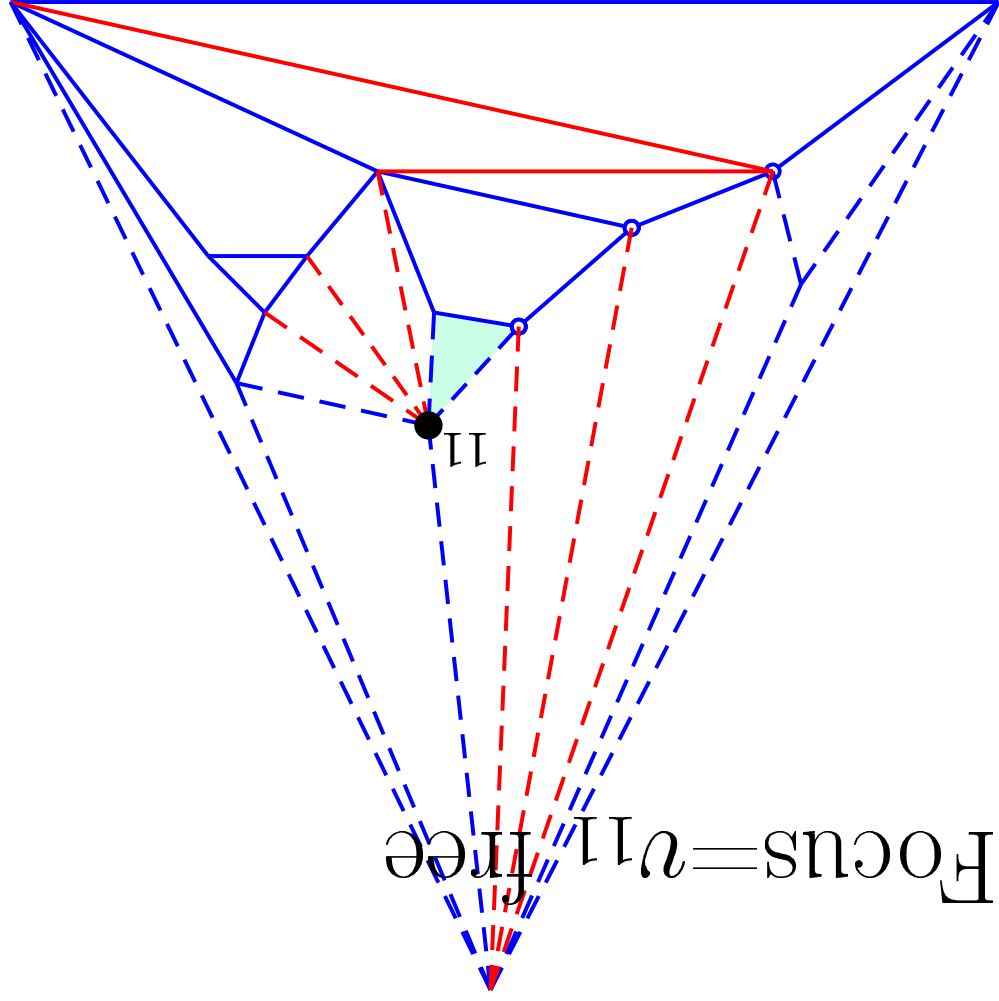
Calcul de la triangulation

Focus = v_{11}



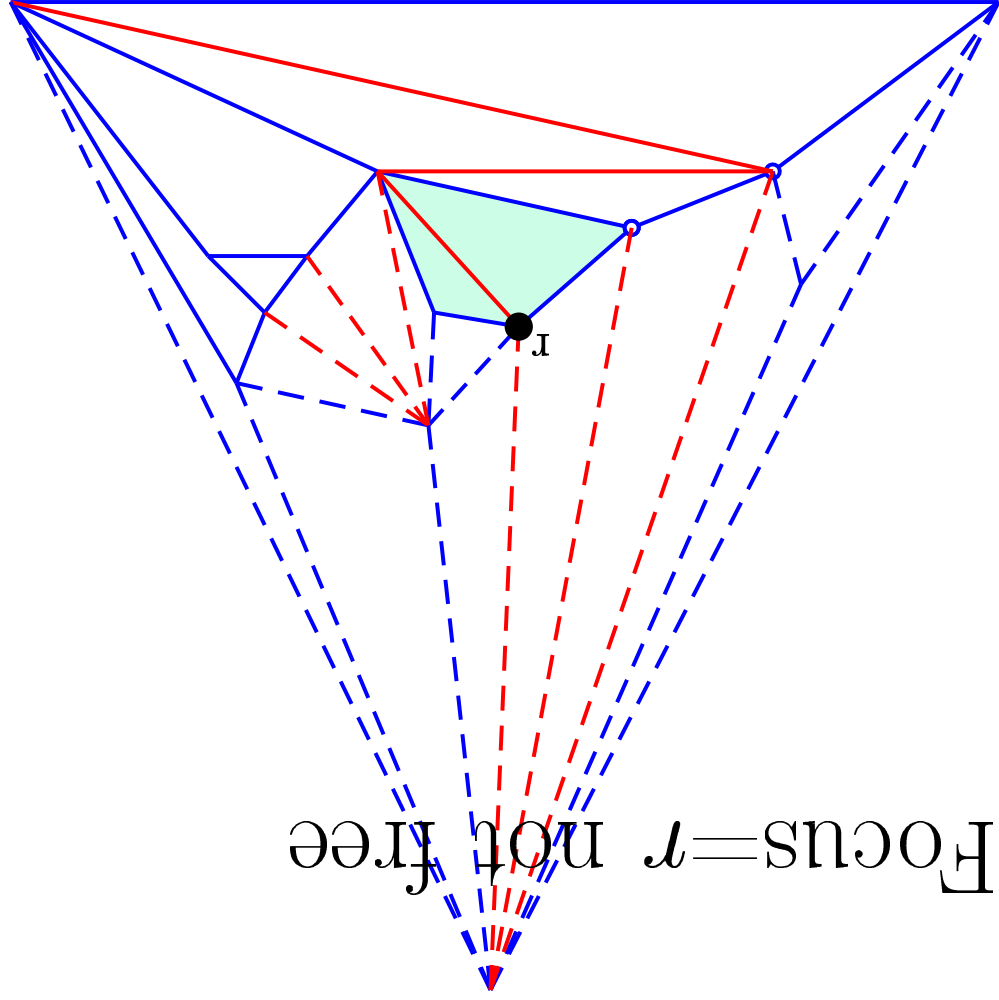
Calcul de la triangulation

Focus = v_{11}



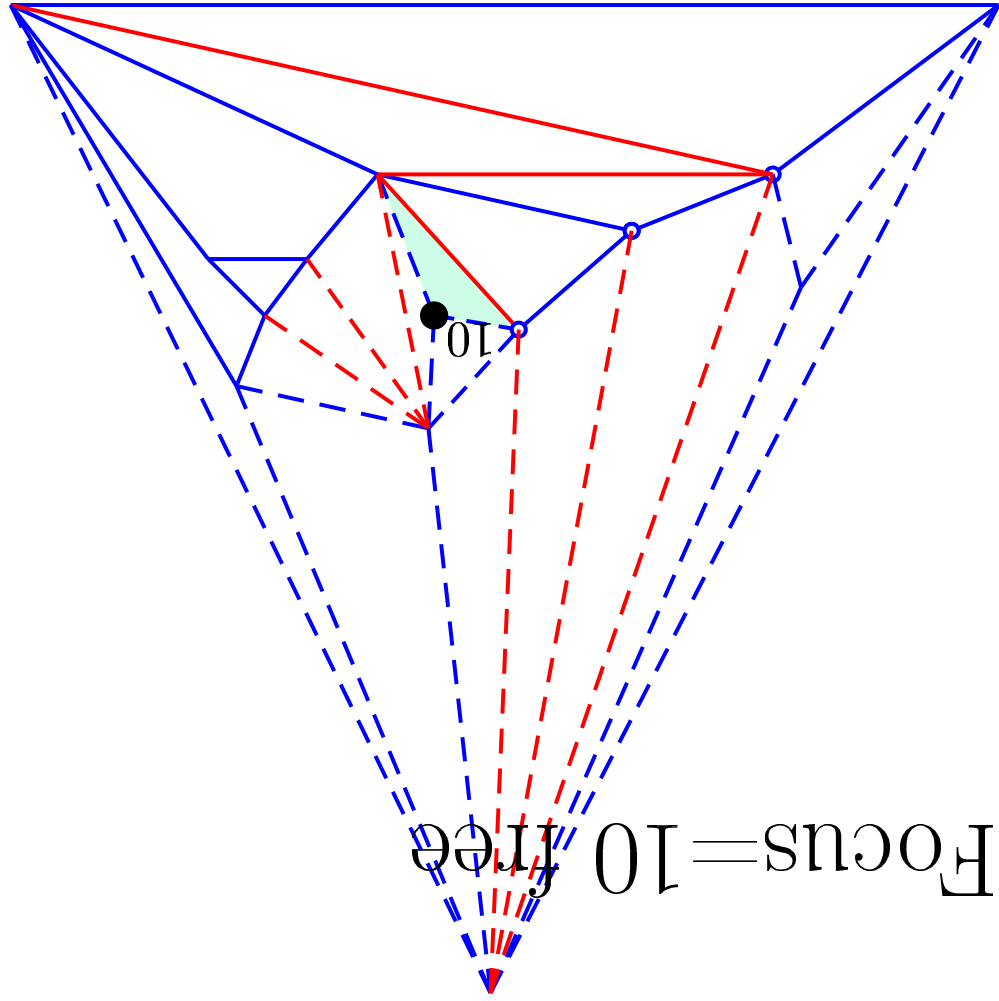
Calcul de la triangulation

Focus = r not free



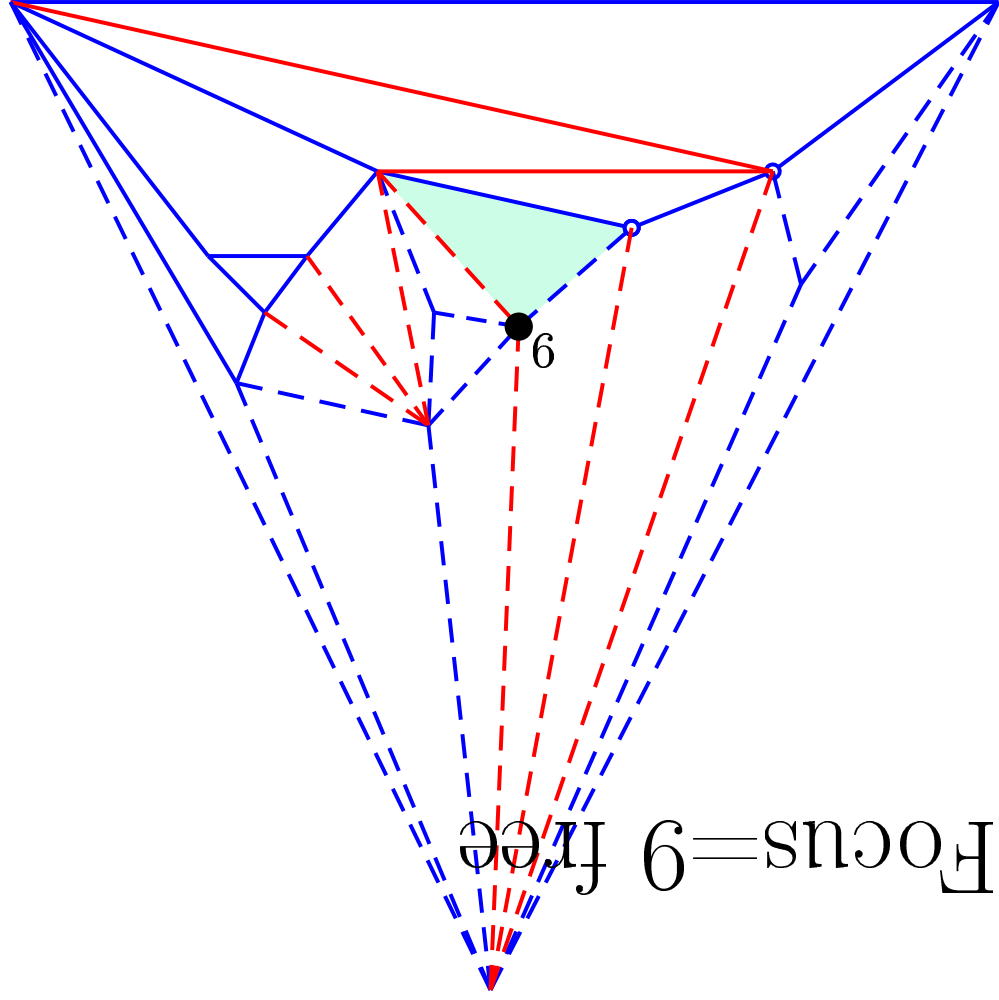
Calcul de la triangulation

Focus = 10 free



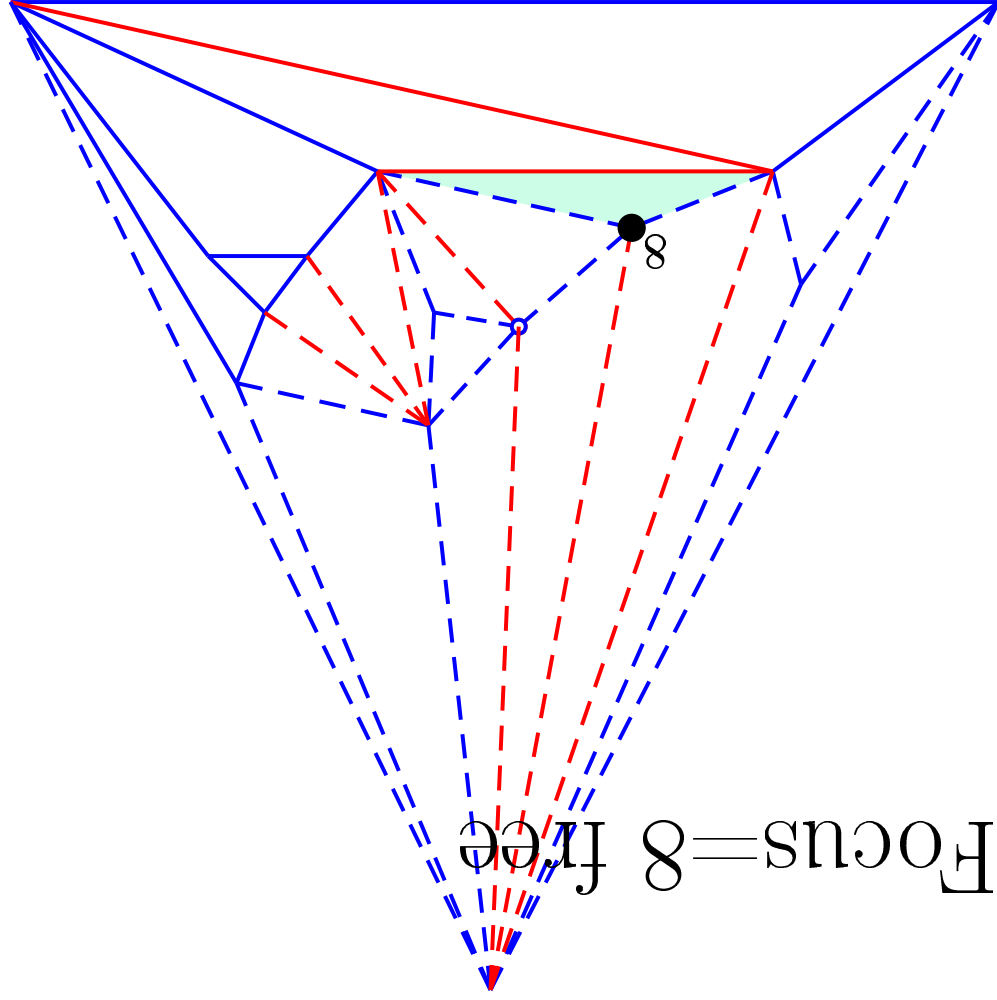
Calcul de la triangulation

Focus=9 free



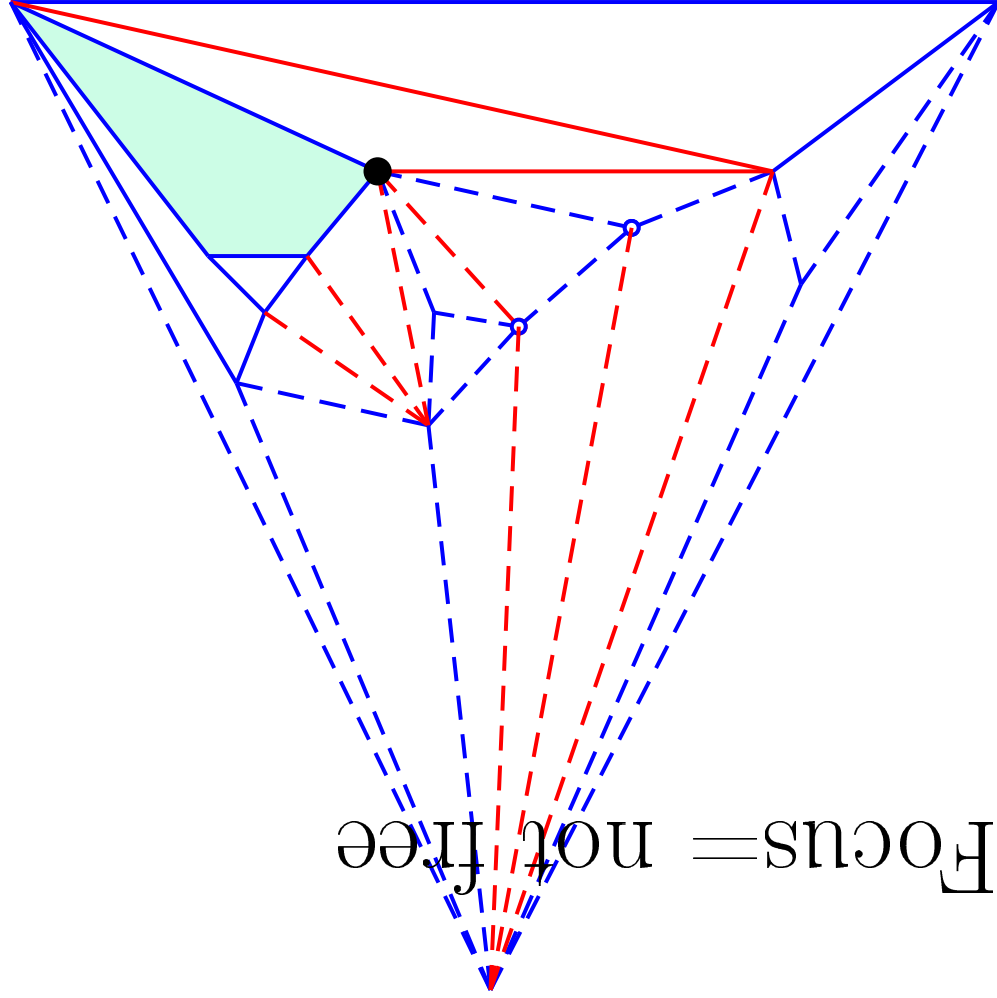
Calcul de la triangulation

Focus = 8 free



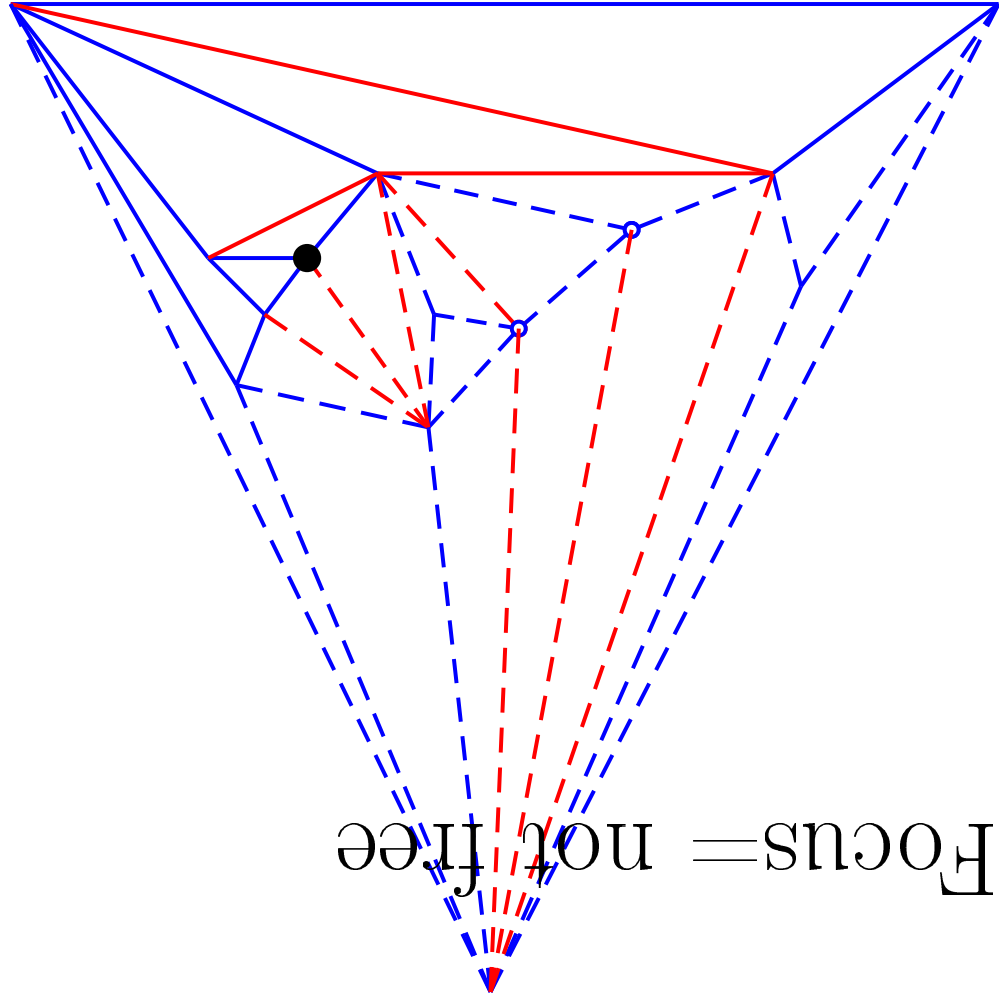
Calcul de la triangulation

Focus = not free

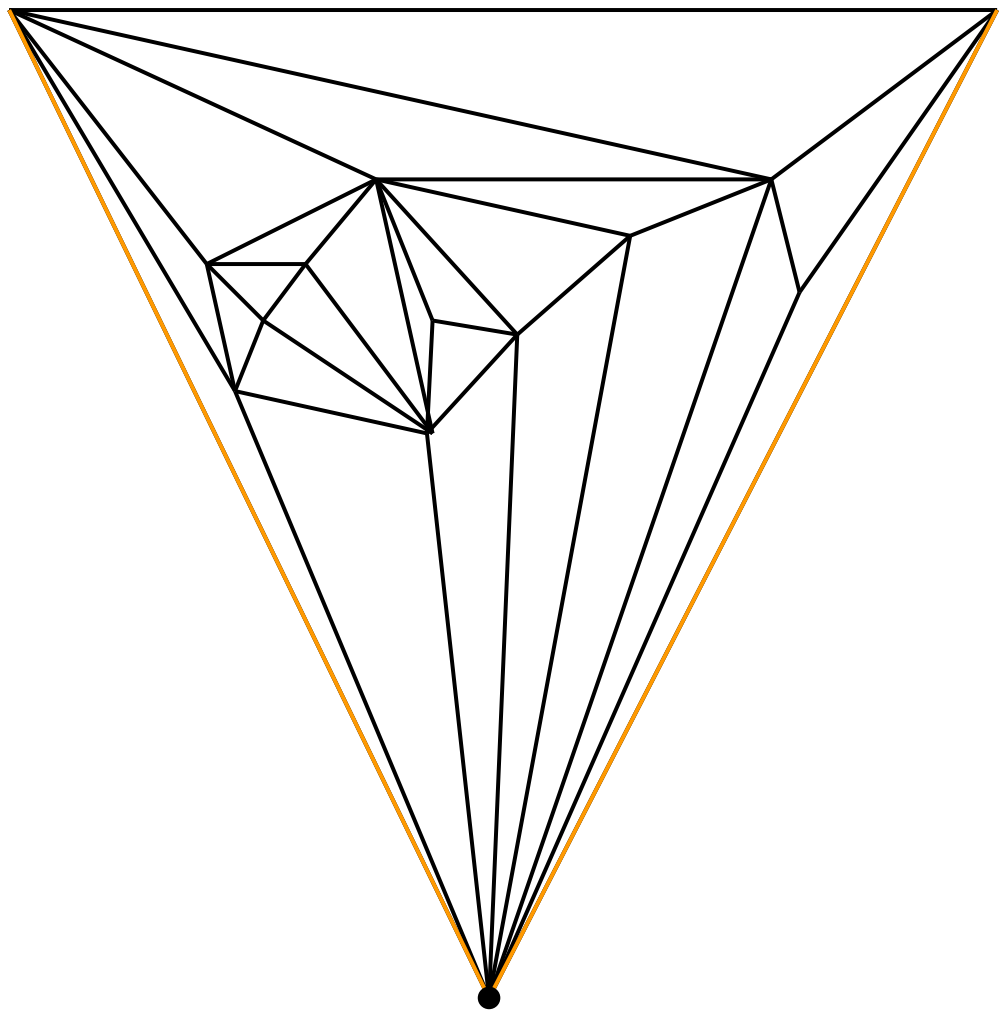


Calcul de la triangulation

Focus = not free

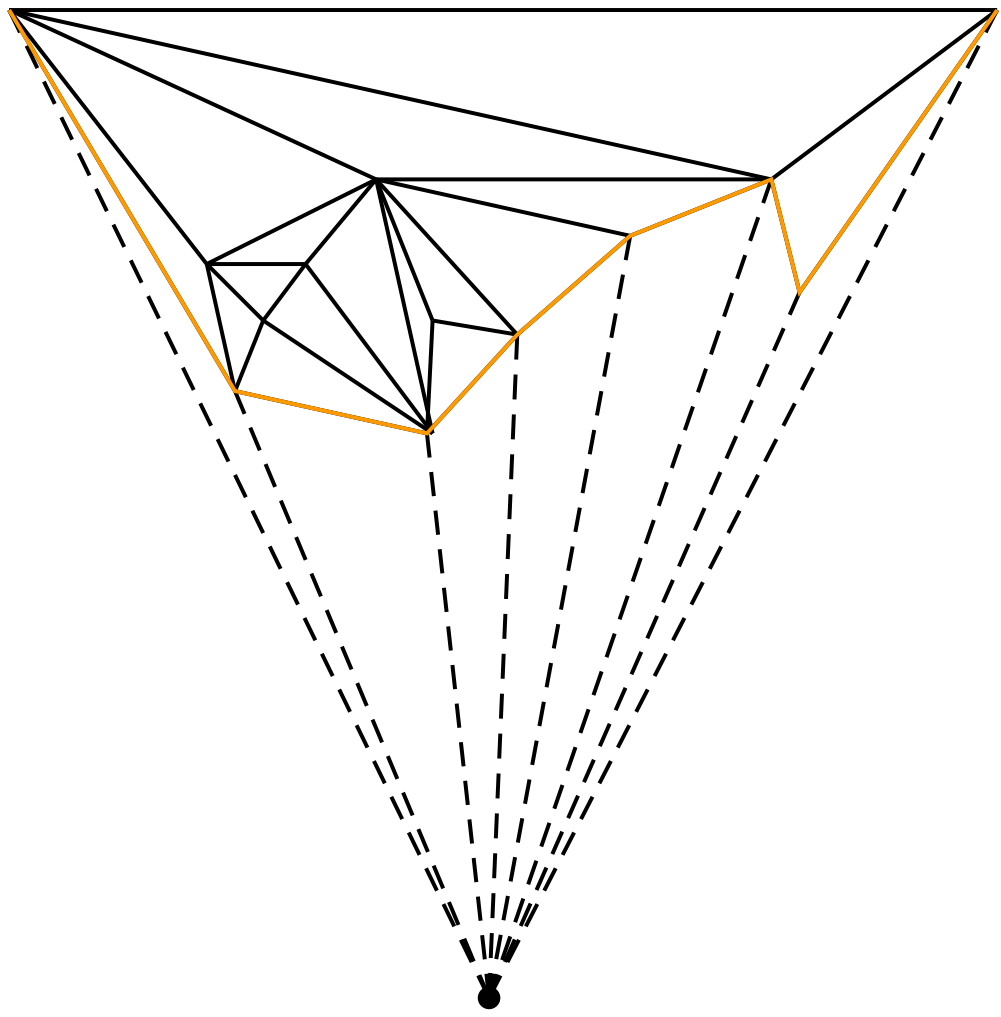


Start decoding



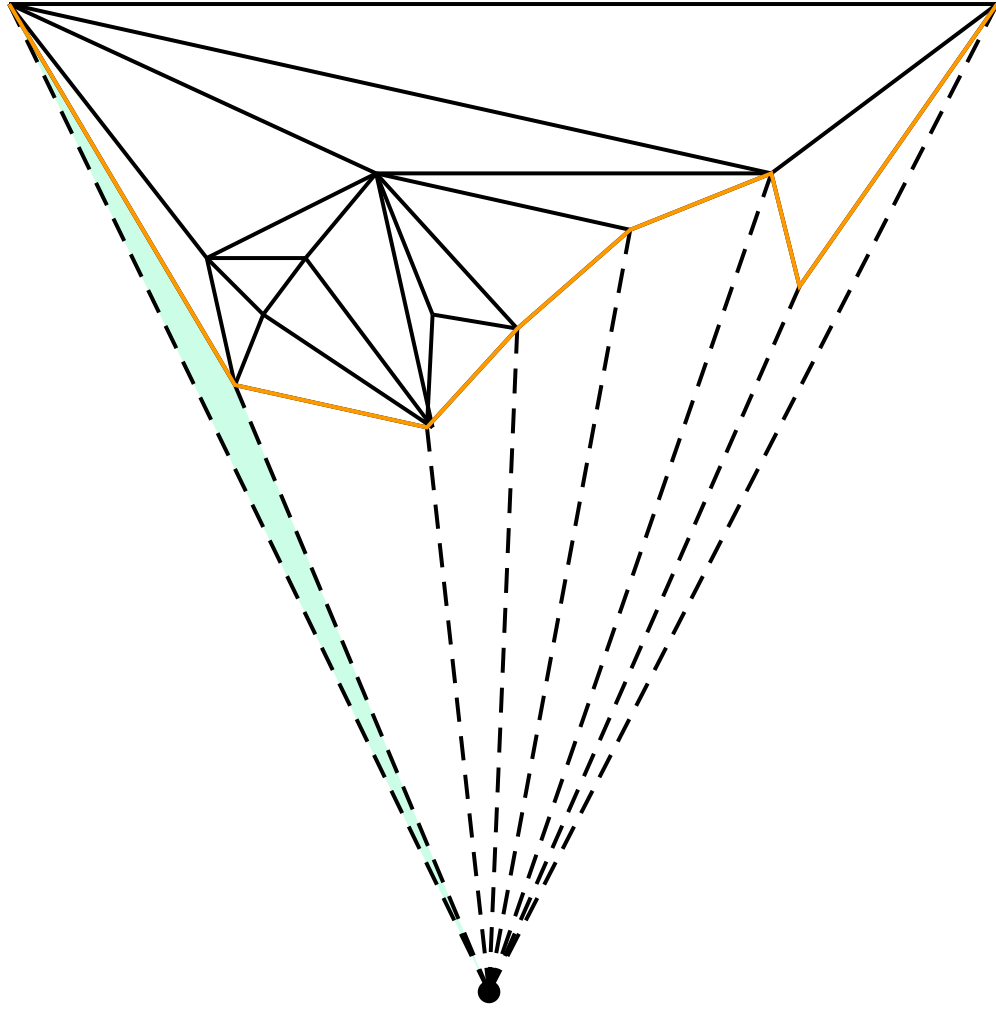
degrees of the faces: 3, 3, 6, 3, 3, 6, 3, 3, 6, 3, 4, 5, 4, 3, 4

Start decoding



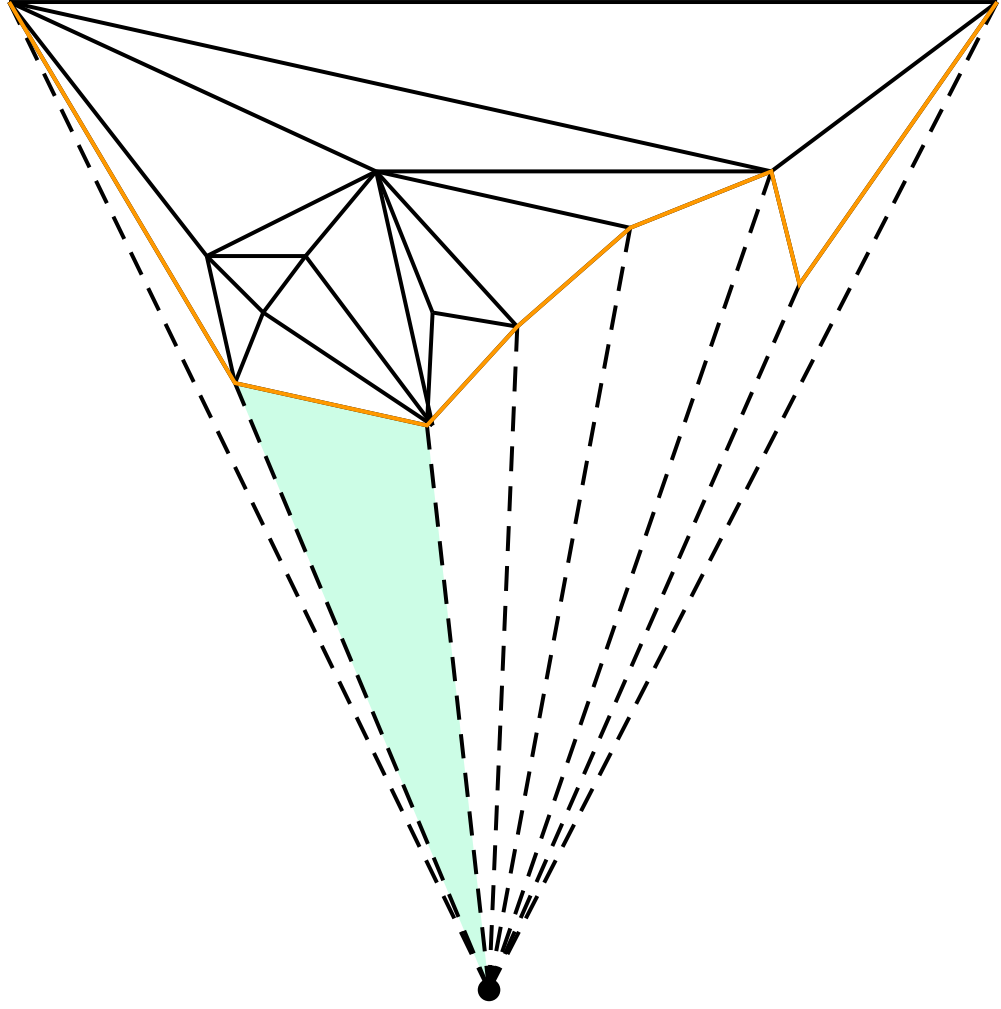
degrees of the faces: 3, 3, 6, 3, 3, 6, 3, 3, 6, 3, 4, 5, 4, 3, 4

Start decoding



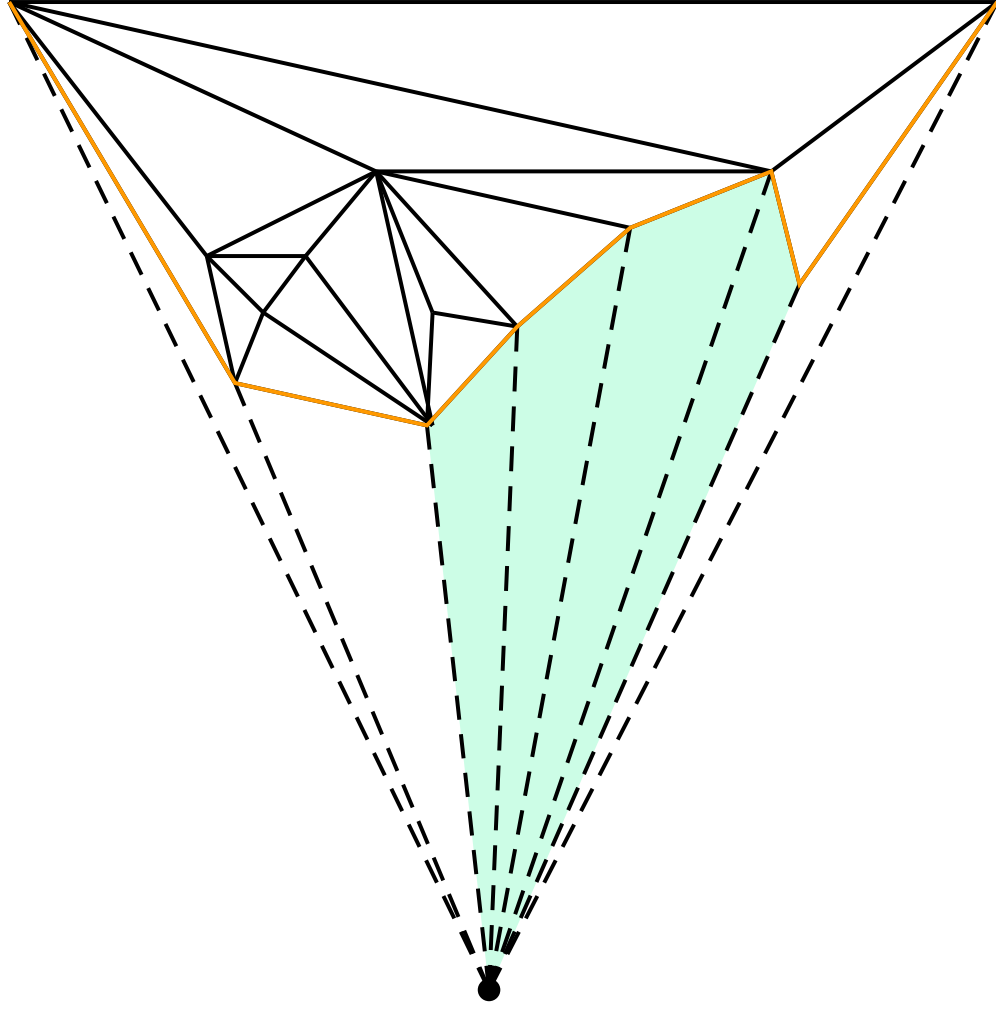
degrees of the faces: $\overline{3}$, 3, 6, 3, 3, 6, 3, 4, 5, 4, 3, 4

Start decoding



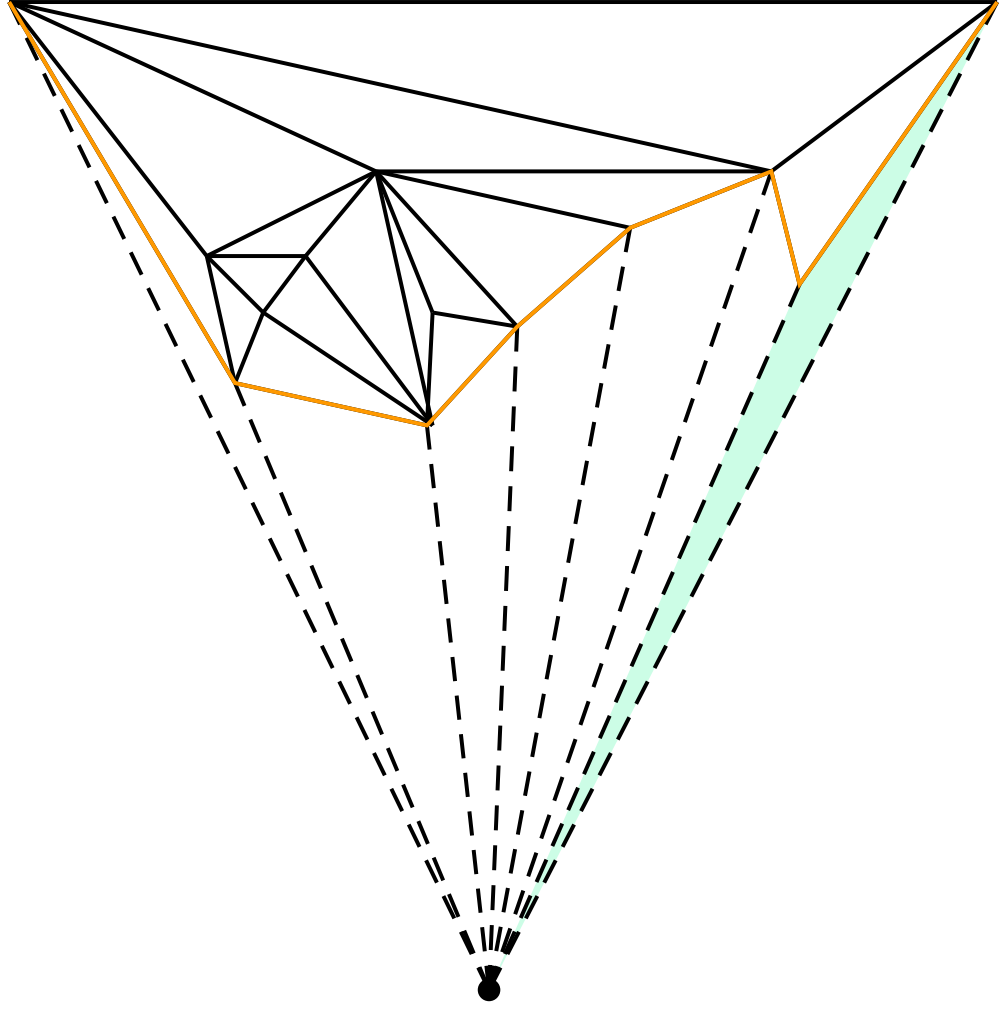
degrees of the faces: $\boxed{3}$, 6, 3, 3, 6, 3, 4, 5, 4, 3, 4

Start decoding



degrees of the faces: $3, 3$ $\overline{6}$, $3, 3, 6, 3, 4, 5, 4, 3, 4$

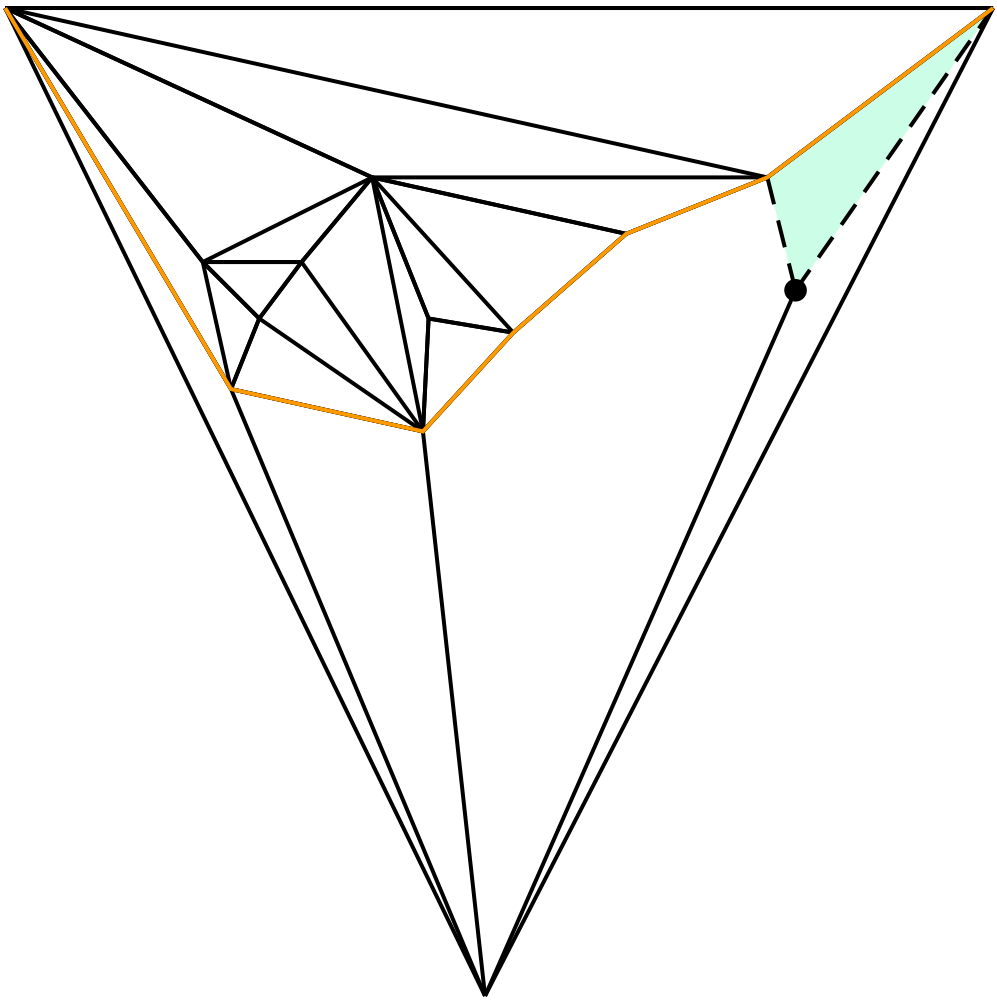
Start decoding



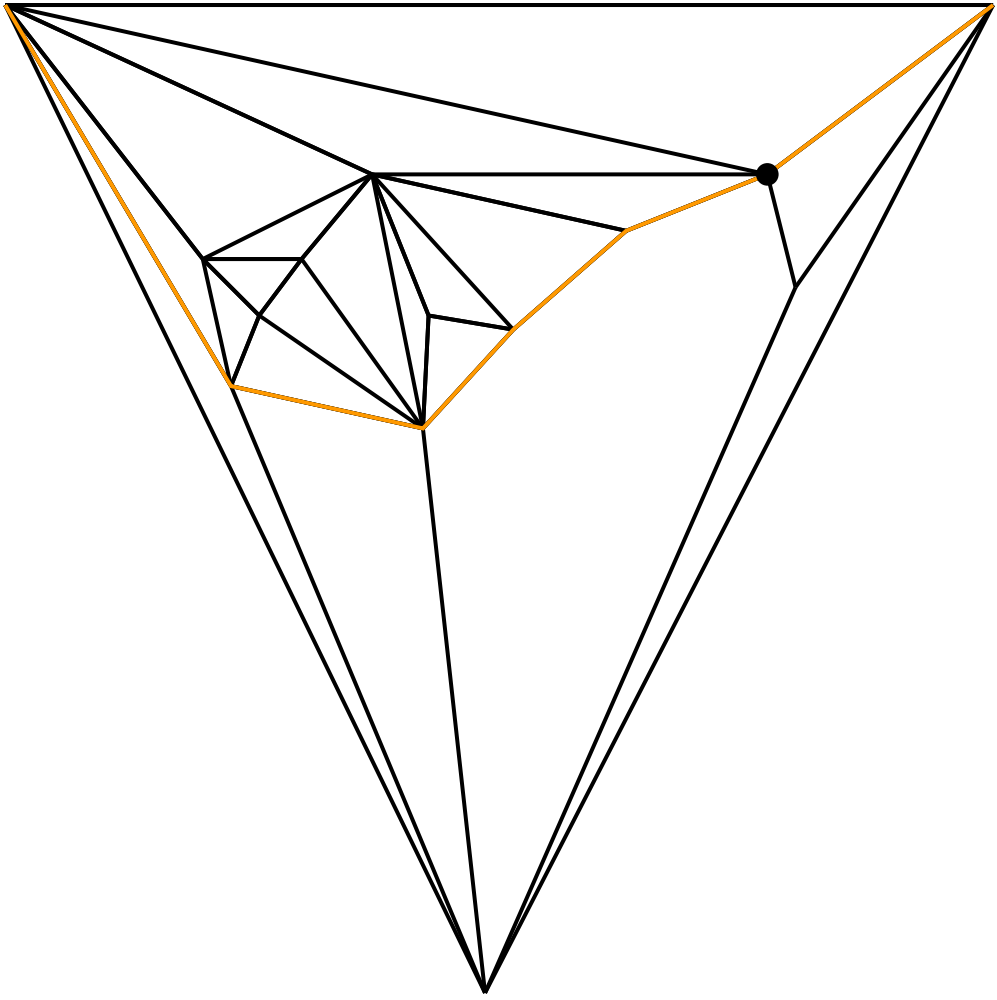
degrees of the faces: $3, 3, 6$ $\boxed{3}$, $3, 6, 3, 4, 5, 4, 3, 4$

decoding

degrees of the faces: 3, 3, 6, 3, $\boxed{3}$, 6, 3, 4, 5, 4, 3, 4



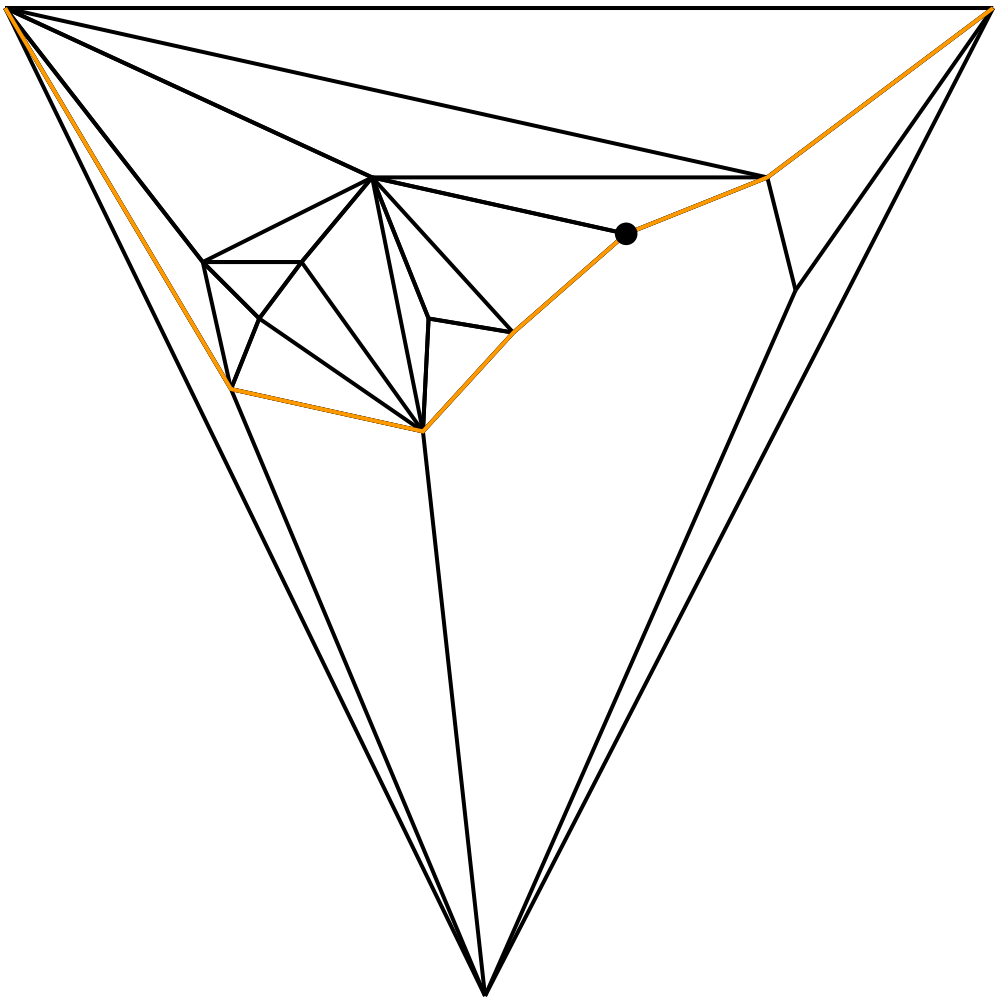
decoding



degrees of the faces: 3, 3, 6, 3, $\boxed{3}$, 6, 3, 4, 5, 4, 3, 4

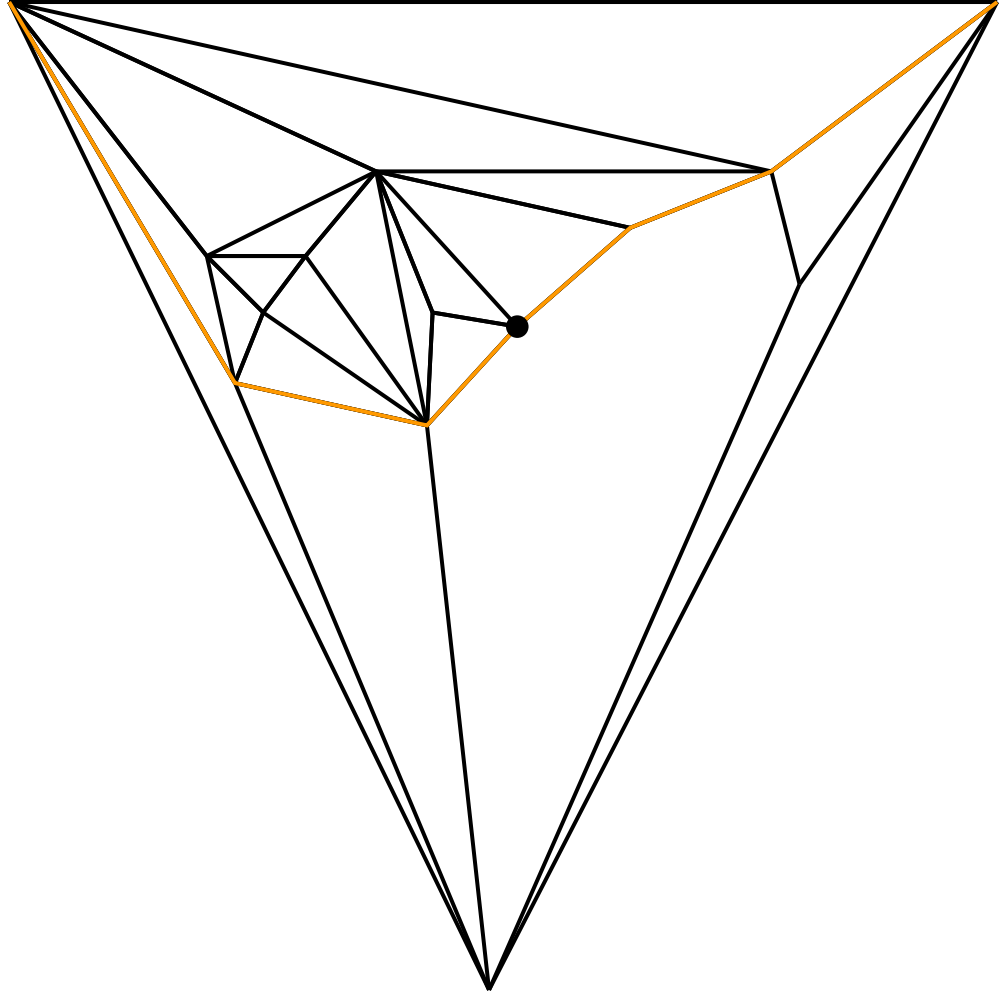
decoding

degrees of the faces: $3, 3, 6, 3, \boxed{3}, 6, 3, 4, 5, 4, 3, 4$

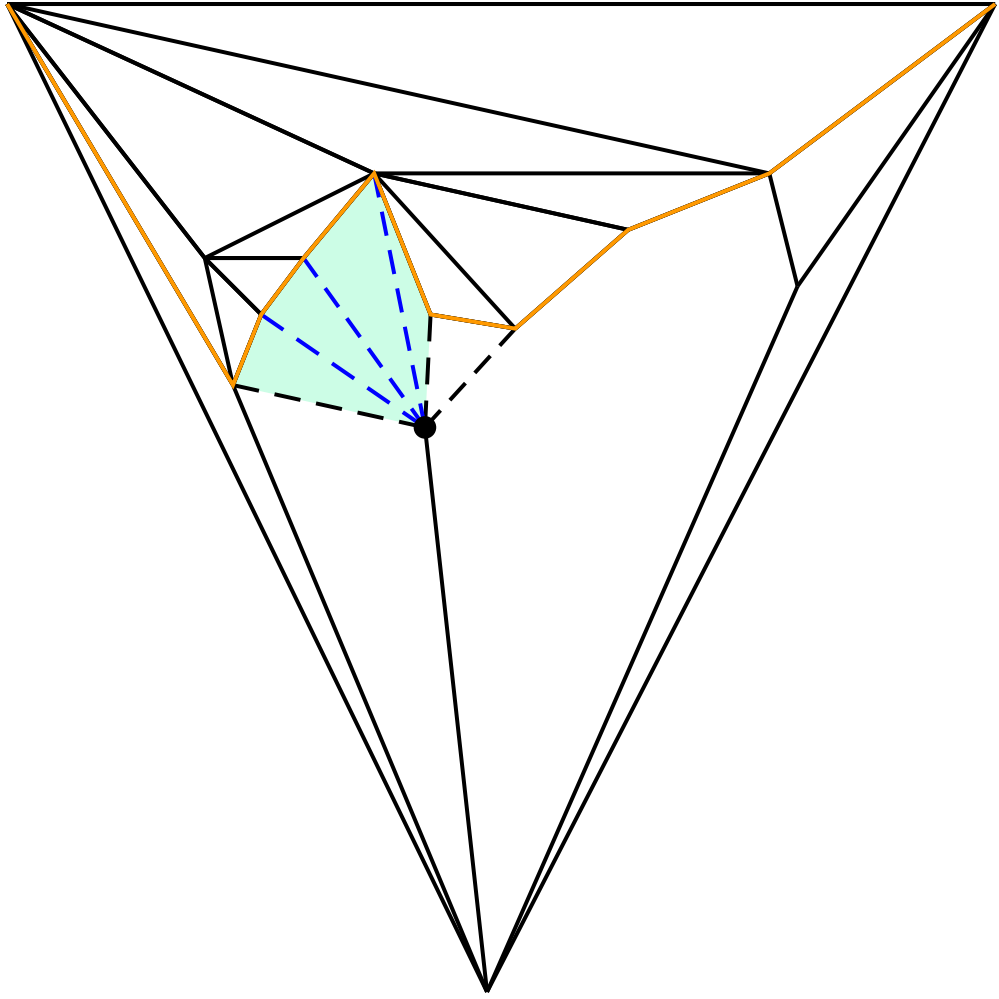


decoding

degrees of the faces: $3, 3, 6, 3, \boxed{3}, 6, 3, 4, 5, 4, 3, 4$



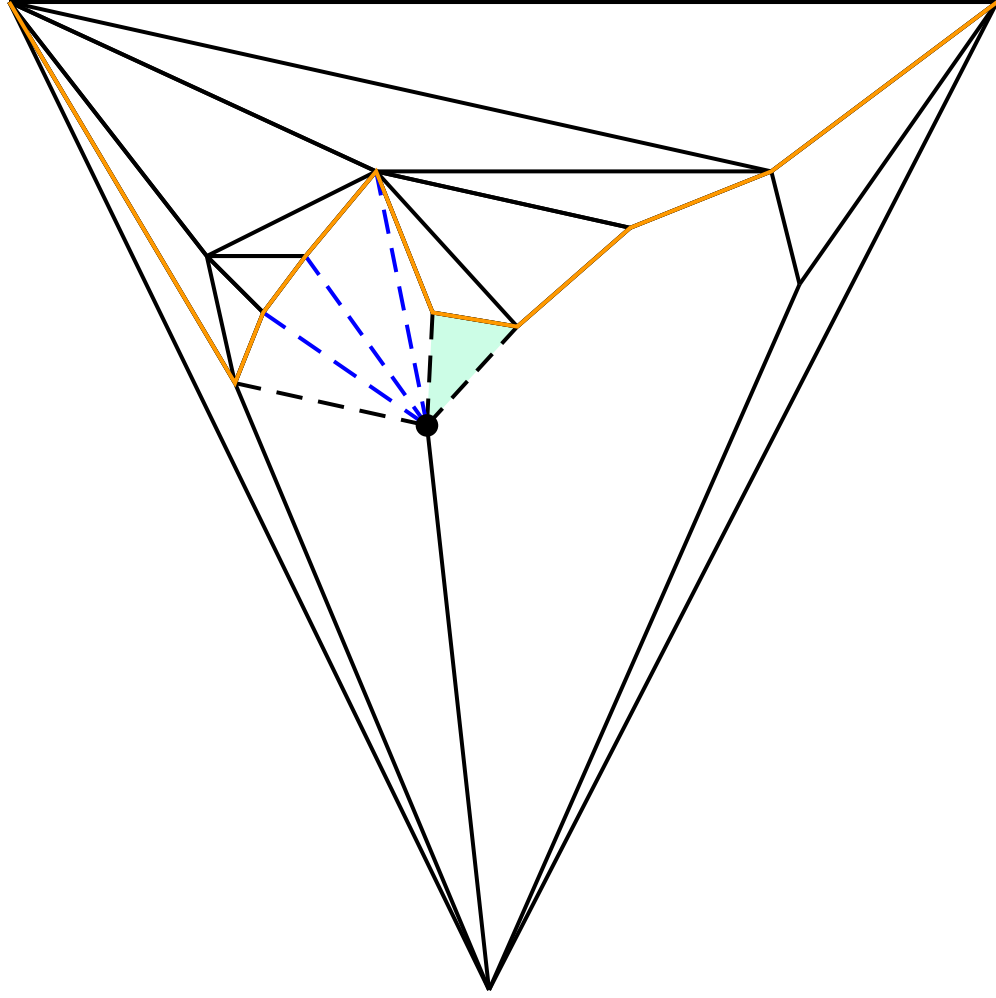
decoding



degrees of the faces: 3, 3, 6, 3, 3, $\overline{6}$, 3, 4, 5, 4, 3, 4

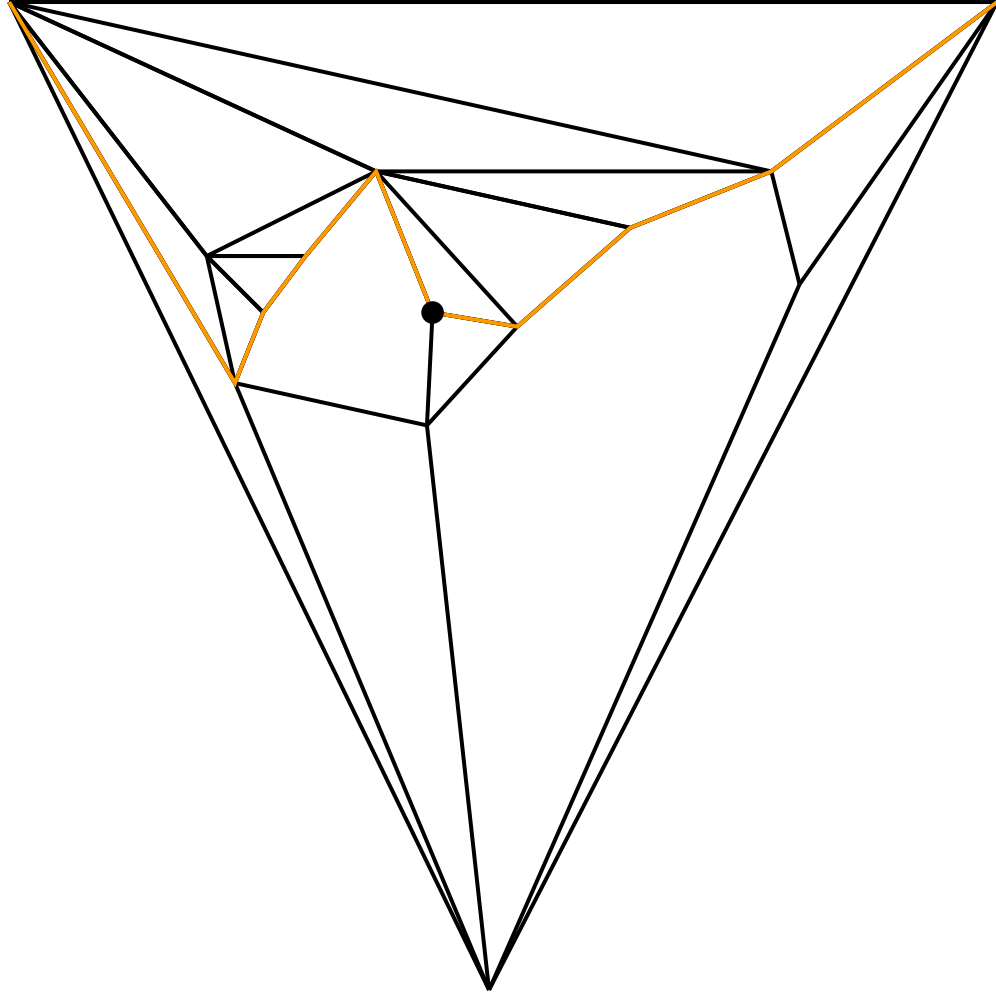
decoding

degrees of the faces: $3, 3, 6, 3, 3, 6, \boxed{3}, 4, 5, 4, 3, 4$



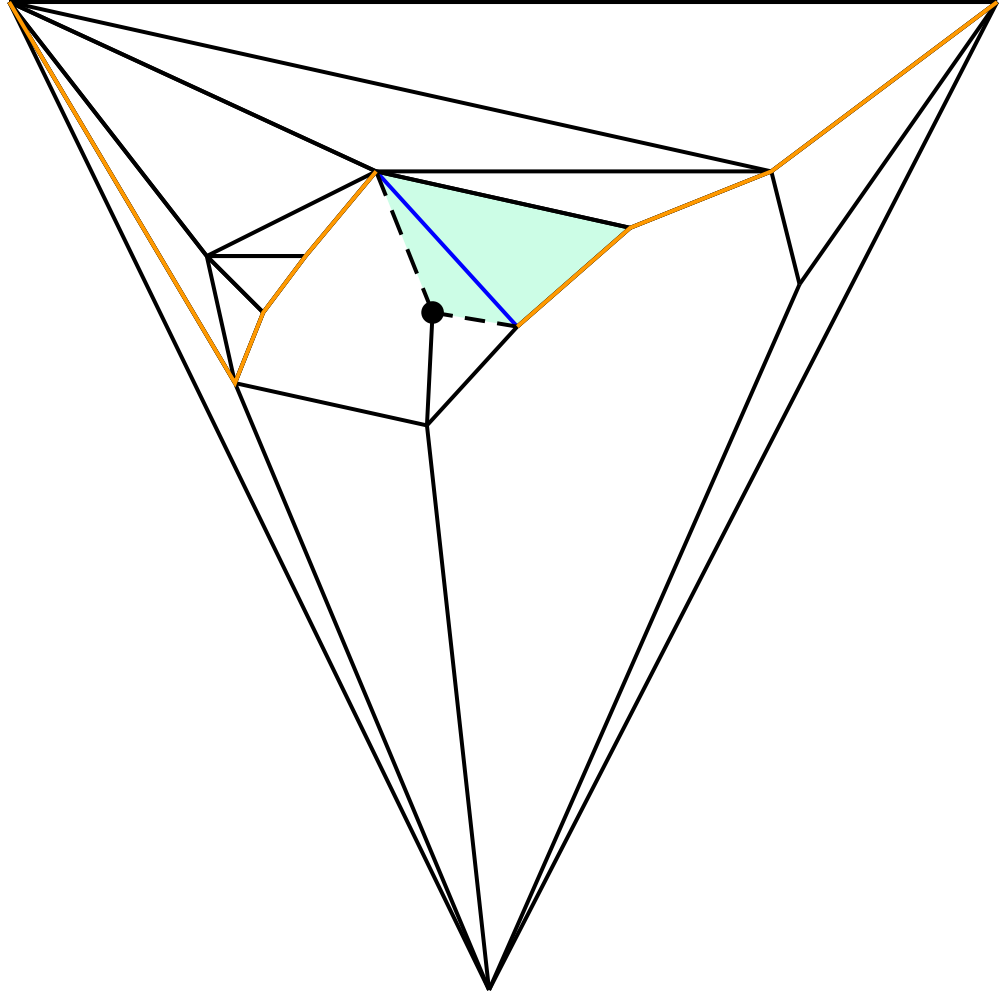
decoding

degrees of the faces: $3, 3, 6, 3, 3, 6, \boxed{3}, 4, 5, 4, 3, 4$



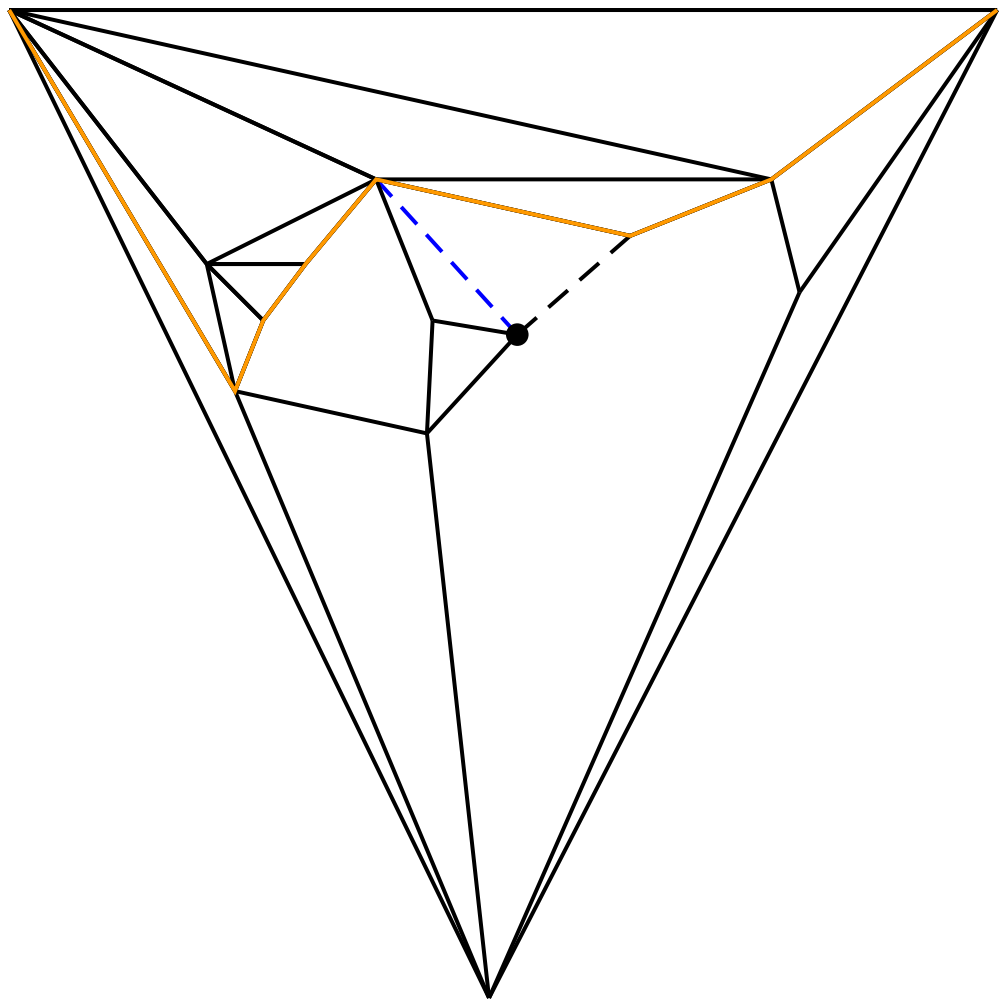
decoding

degrees of the faces: 3, 3, 6, 3, 3, 6, 3, $\boxed{4}$, 5, 4, 3, 4



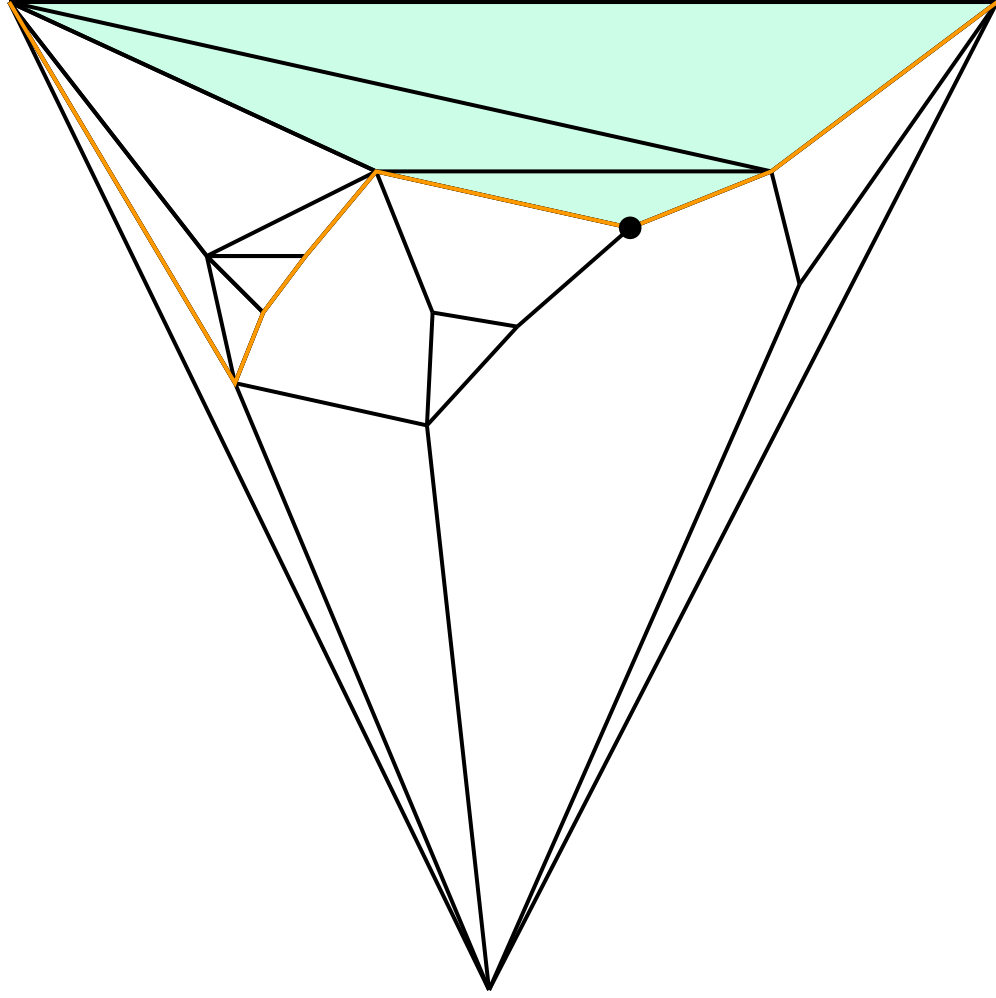
decoding

degrees of the faces: 3, 3, 6, 3, 3, 6, 3, $\boxed{4}$, 5, 4, 3, 4



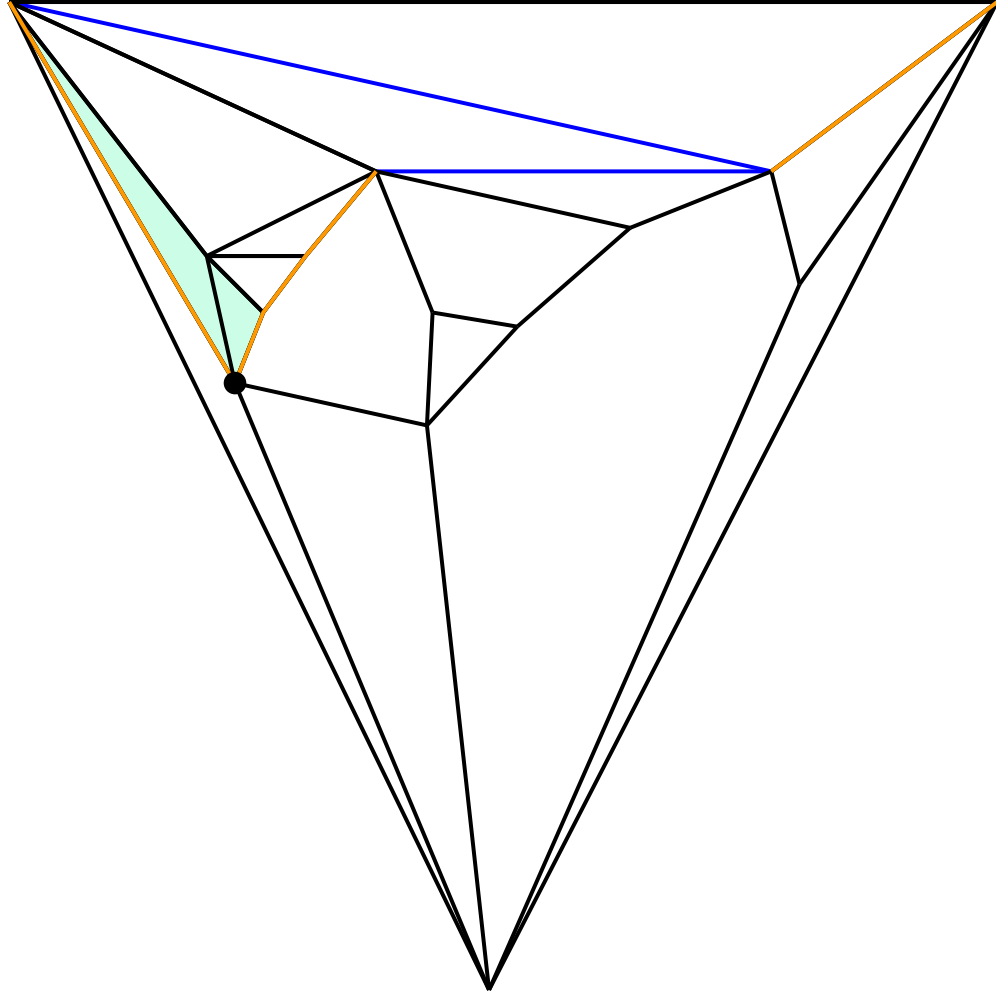
decoding

degrees of the faces: $3, 3, 6, 3, 3, 6, 3, 4$ \square $\bar{5}$, $4, 3, 4$



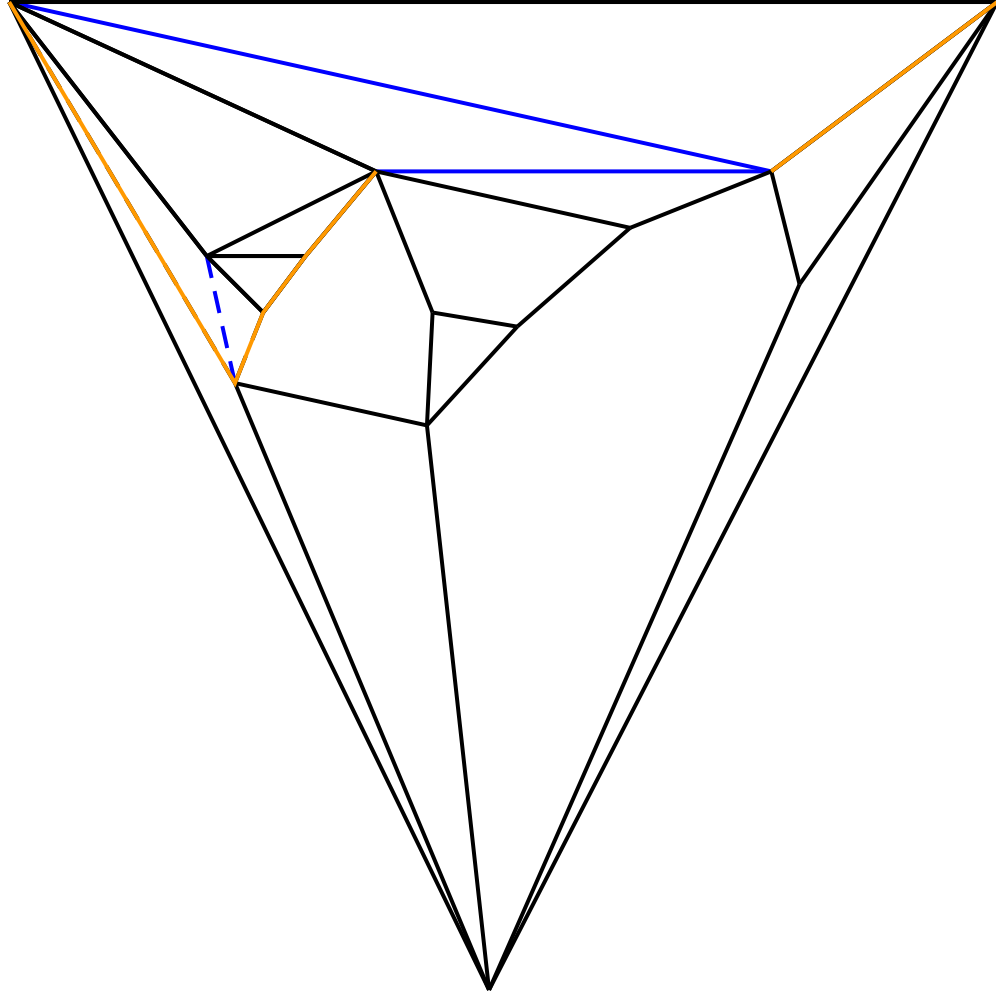
decoding

degrees of the faces: $3, 3, 6, 3, 3, 6, 3, 3, 4, 5$ $\overline{4}$, $3, 4$

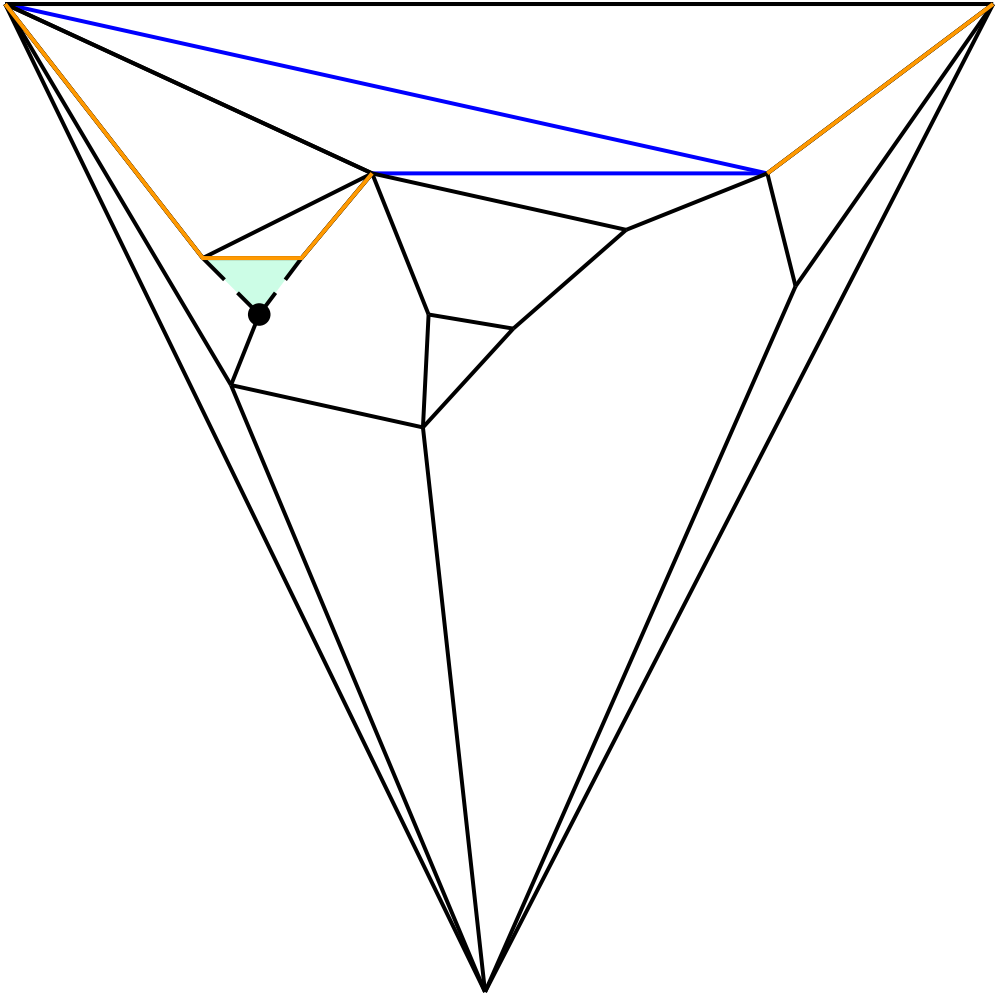


decoding

degrees of the faces: $3, 3, 6, 3, 3, 6, 3, 3, 4, 5$ $\overline{4}$, $3, 4$

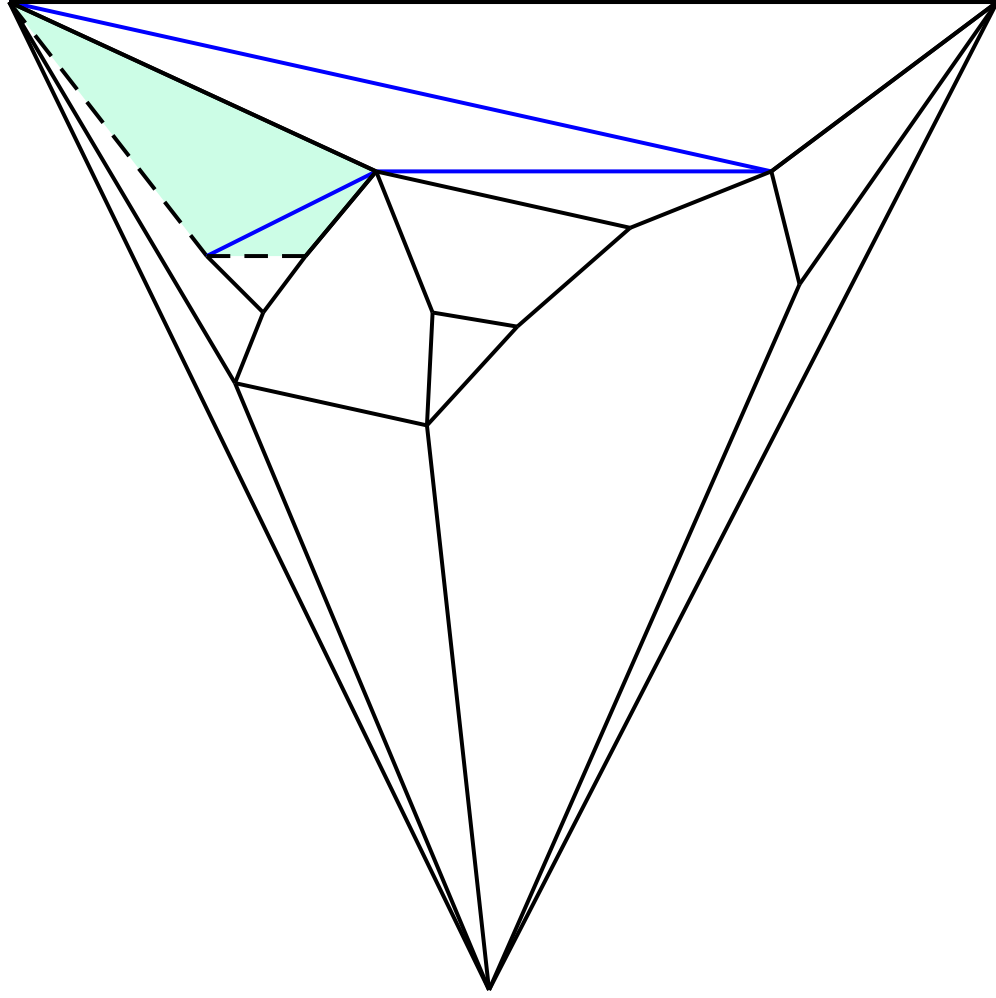


decoding



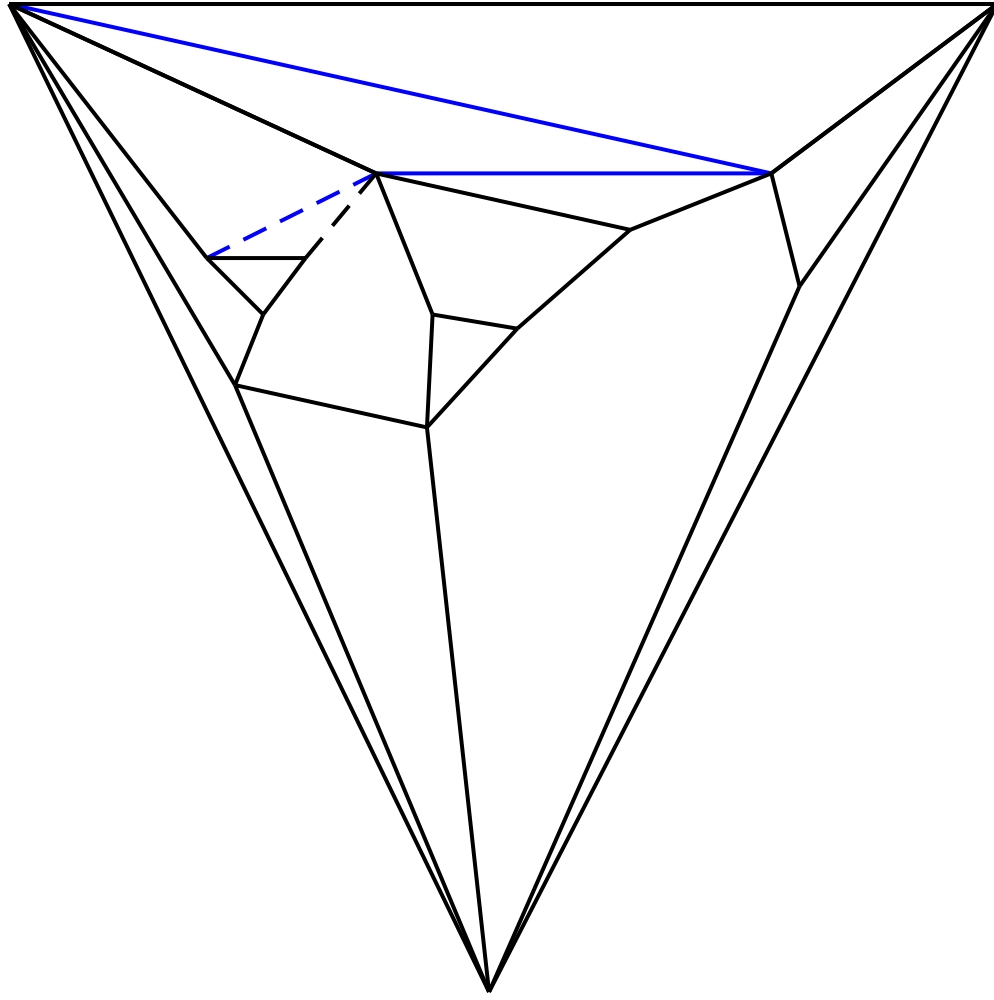
degrees of the faces: 3, 3, 6, 3, 3, 6, 3, 3, 4, 5, 4, $\overline{3}$, 4

decoding

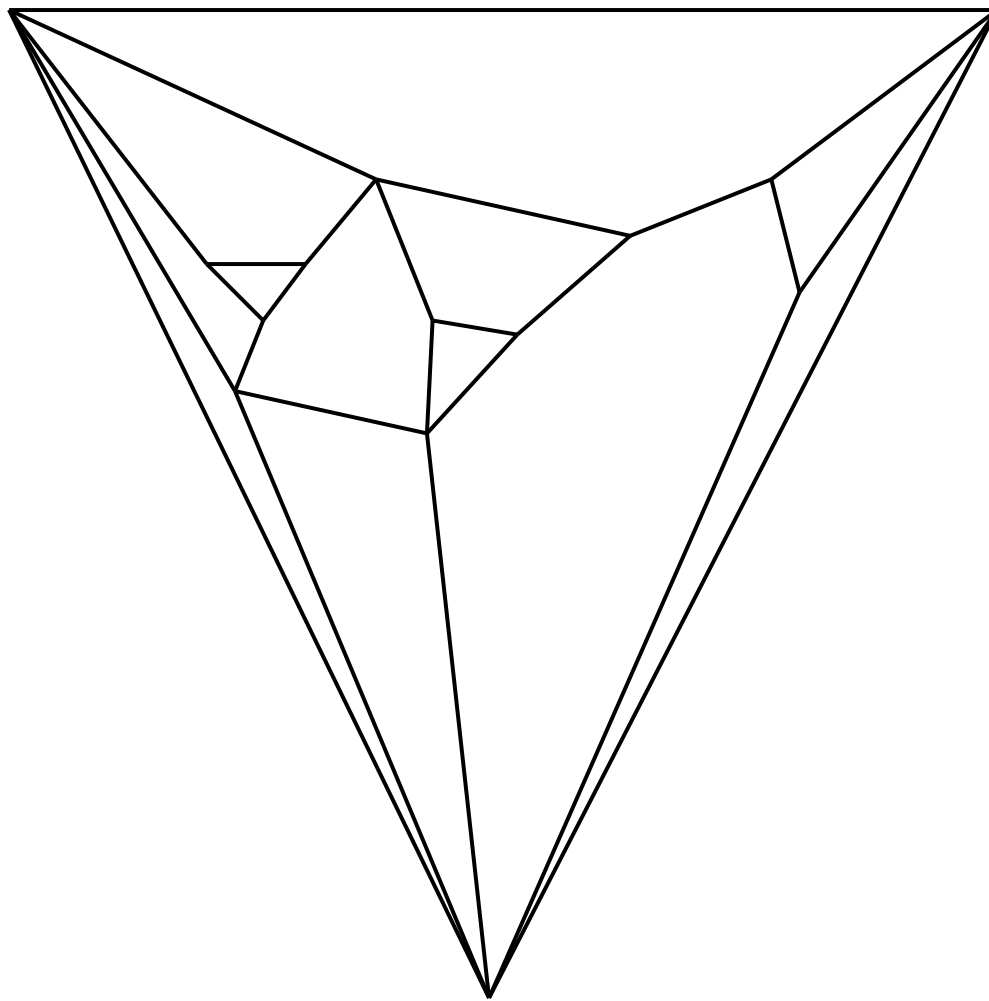


degrees of the faces: 3, 3, 6, 3, 3, 6, 3, 3, 4, 5, 4, 3 $\bar{4}$

decoding



degrees of the faces: 3, 3, 6, 3, 3, 6, 3, 3, 6, 3, 4, 5, 4, 3, 4



decoding

Coût de la Detriangulation

Lemme 1. Soit T_G la triangulation canonique d'un graphe G , et Σ_G la suite des degrés des faces de G générée par l'algorithme de triangulation. Alors on peut reconstruire G à partir de Σ_G avec au plus

$$m + o(m) \text{ bits}$$

- évaluation de l'entropie (maillages polygonaux) (Khodakovski et al. '01)

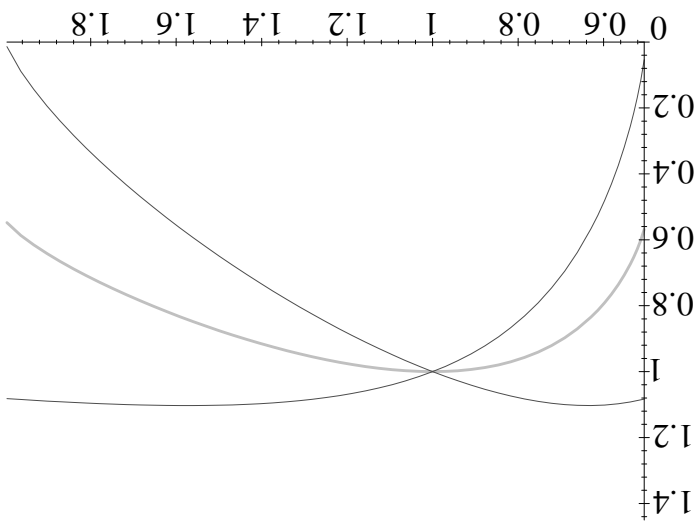
$$-\left(\sum_{i>2} p_i \log p_i + \sum_{j>2} q_j \log q_j\right) \leq 2 \text{ bits/edge}$$

Le maximum 2 est atteint pour $r = \frac{V}{E} = 1$

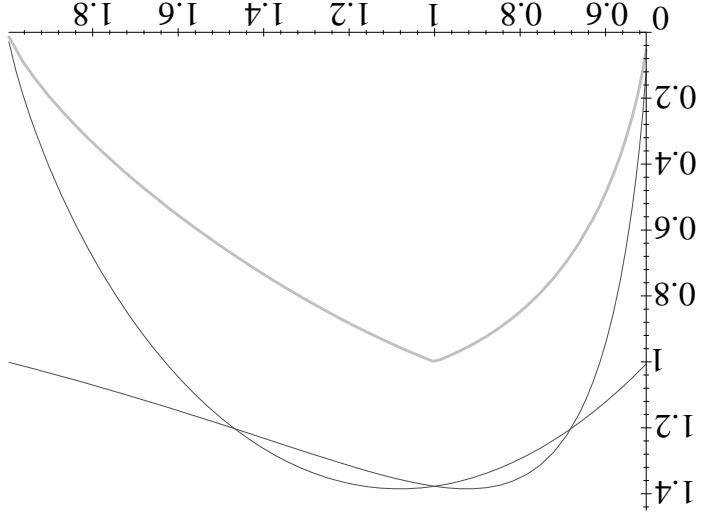
$$\mathcal{E}_* := \sum_{i \geq 2} p_i \log 1/p_i \quad \text{et} \quad \mathcal{E} := \sum_{j \geq 2} q_j \log 1/q_j$$

$\mathcal{E}, \mathcal{E}_*$: entropies des deux suites des degrés des faces et des sommets :

$$\min \{ f \cdot \mathcal{E}, n \cdot \mathcal{E}_* \} \leq \frac{2}{f \cdot \mathcal{E} + n \cdot \mathcal{E}_*} = \frac{e+2}{2} \left(\frac{1}{r+1} \mathcal{E} + \frac{r+1}{r} \mathcal{E}_* \right) \leq e+o(e) \text{ bits}$$



Comparison avec une méthode triviale



Taux de compression de notre algorithme: $\min(f \cdot \mathcal{E}, n \cdot \mathcal{E}^*)$

Borne supérieure pour le taux de compression

Théorème. *Un graphe planaire simple 3-connexe avec n sommets, m arêtes et f faces peut être codé avec au plus*

$$(3.2451 + 2) \min\{n, f\} + 1 = 2.62e + o(e)$$

- If G is 3-connected then so is its dual G^* ;
- if $n > 3$ ($f > 3$) then G^* is simple;
- $\|code(G, G^*)\| \leq (3.2451 + 2) \min\{n, f\}$ bits.

$$\frac{(3.2451 + 2)}{2} (e + 1) + 1 = 2.62e + o(e)$$

Résultats

3D model	V	F	E	G bpe	G* bpe
triceratops	2832	2834	5664	2.14	1.65
cupie	2984	3032	6004	2.07	2.15
lion head	2109	2312	4419	2.03	2.16
shark	2560	2562	5120	1.98	1.62
Beethoven	2643	2816	5438	2.17	2.32

3D model	E	our+TG (b/e)	our+AD (b/e)	IS (b/e)
triceratops	5664	1.36	1.22	0.59
lion head	4419	1.61	1.45	0.87
Beethoven	5438	1.61	1.38	0.99

Changement de l'entropie des degrés des sommets après triangulation

3D model	G b/v	G^* b/v	T_G b/v	T_{G^*} b/v	G b/e	G^* b/e
triceratops	0.06	1.04	0.92	1.16	0.52	0.03
lion head	0.89	1.02	1.04	1.26	0.48	0.47
shark	0.008	0.72	0.73	0.96	0.36	0.004
man	1.05	1.15	1.33	1.26	0.59	0.51
AI	1.16	0.80	1.09	-	0.41	0.57
ear	0.64	0.17	0.96	-	0.08	0.32

- Travaux futurs
- algorithme pour les graphes 3-connexes avec une borne supérieure garantie proche de l'entropie de Tutte
 - ≈ 2 bpe
 - surfaces avec bords;
 - surfaces de genre quelconque;