De la géométrie algorithmique

au calcul géométrique

l'exemple

de la triangulation de Delaunay

La randomisation

Analyse des performances

Borne inférieure
Borne supérieure

Complexité asymptotique

Analyse dans le cas le pire

Analyse des performances

Borne inférieure
Borne supérieure

Complexité asymptotique

Analyse dans le cas le pire

Algorithmes compliqués

Constantes cachées dans les $O()$

Analyse des performances

Borne inférieure
Borne supérieure

Complexité asymptotique

Analyse dans le cas le pire

Algorithmes compliqués

Constantes cachées dans les $O()$

Algorithmes randomisés

Plus simples
Complexité en moyenne

Analyse des performances

Borne inférieure
Borne supérieure

Complexité asymptotique

Analyse dans le cas le pire

Algorithmes compliqués

Constantes cachées dans les $O()$

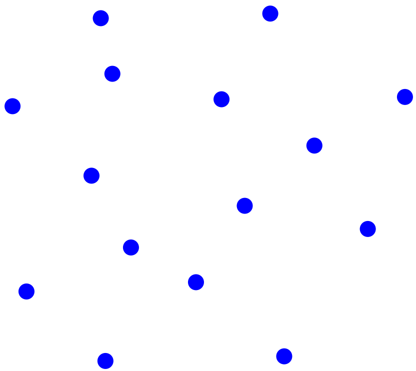
Algorithmes randomisés

Plus simples

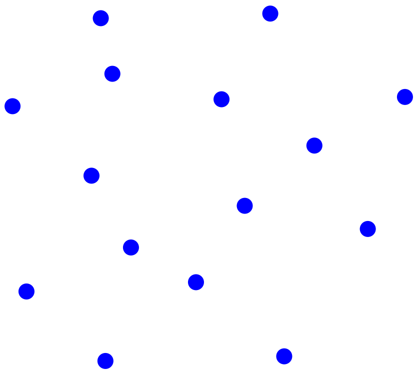
Complexité en moyenne

en moyenne sur quoi ?

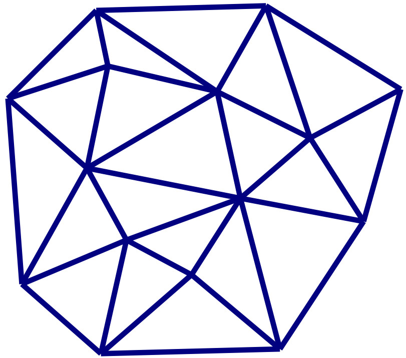
Données

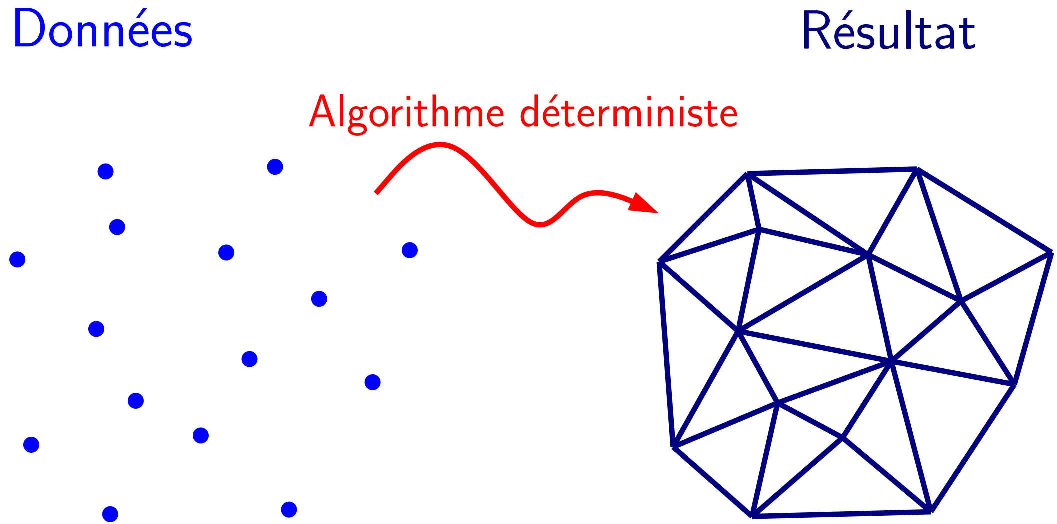


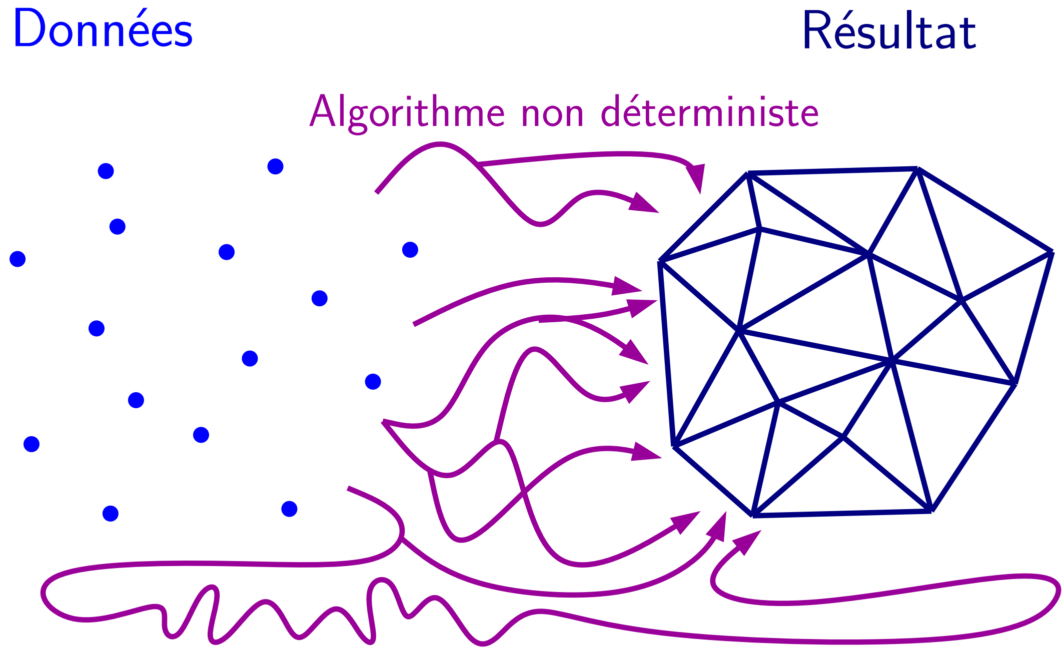
Données

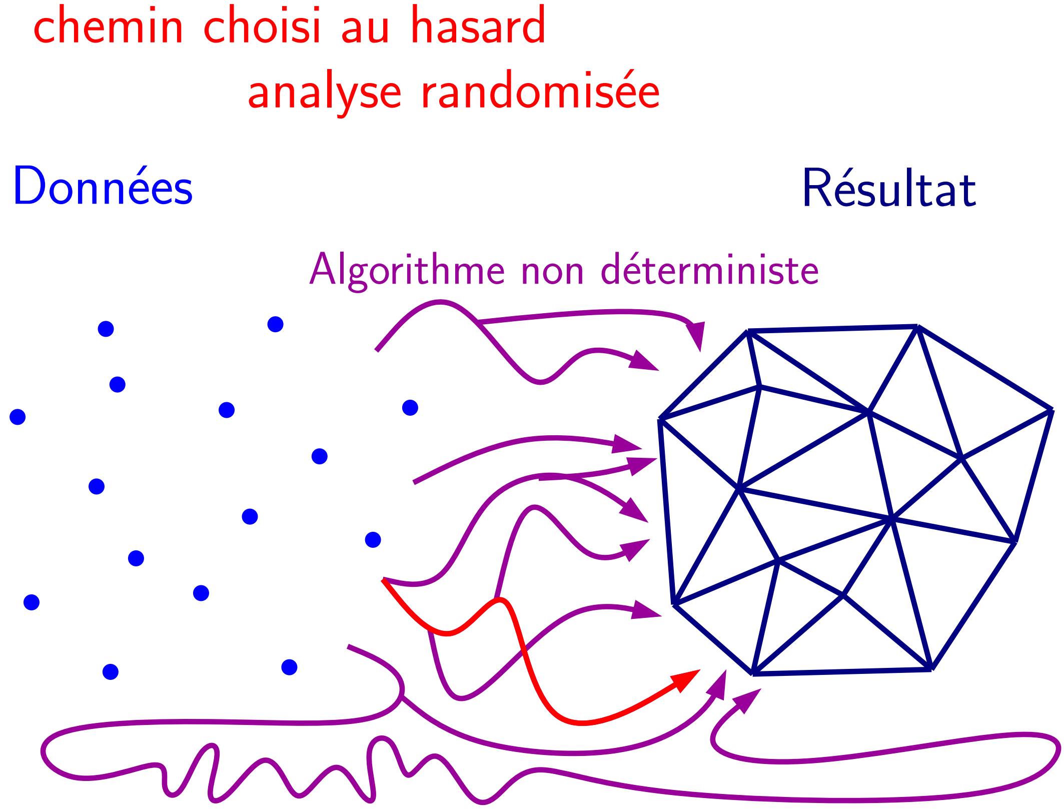


Résultat




Données
Résultat
Algorithme déterministe


Données
Résultat
Algorithme non déterministe


chemin choisi au hasard
analyse randomisée
Données
Résultat
Algorithme non déterministe

Comment introduire du hasard ?

Pas d'hypothèses sur la répartition des données

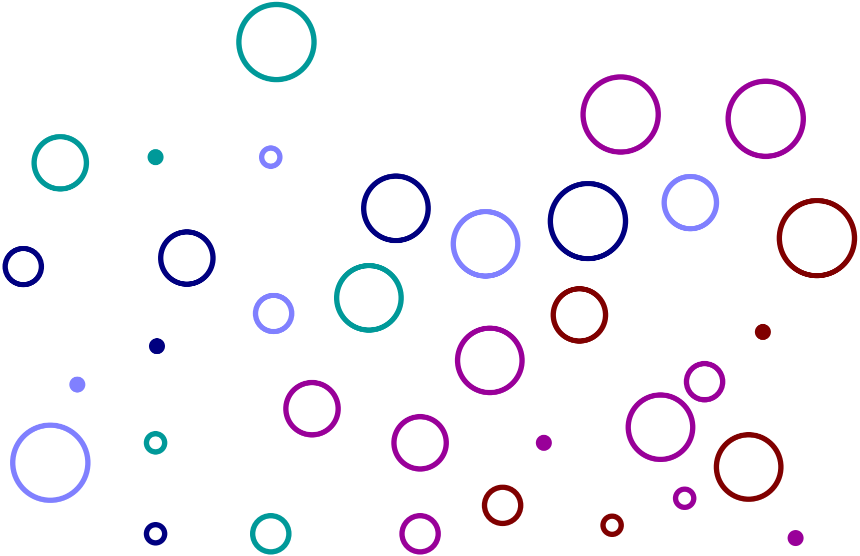
tirage à pile ou face

division fusion (division au hasard)

incrémental (ordre aléatoire)

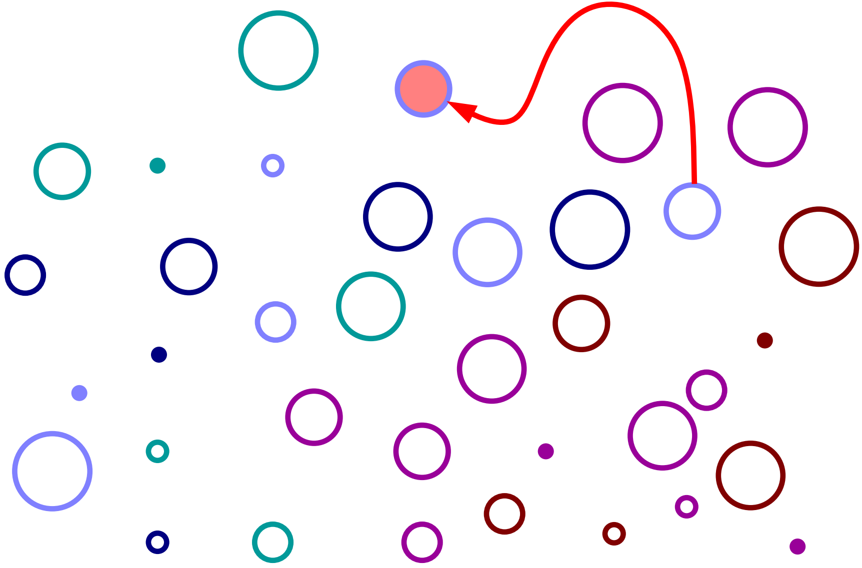
Médian linéaire

Médian linéaire



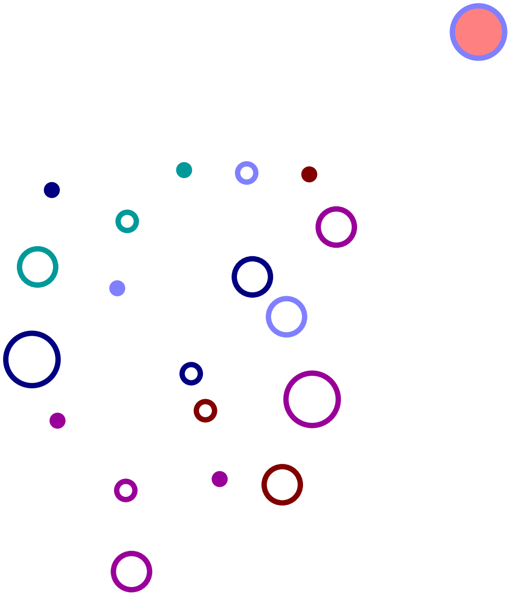
n nombres

Médian linéaire



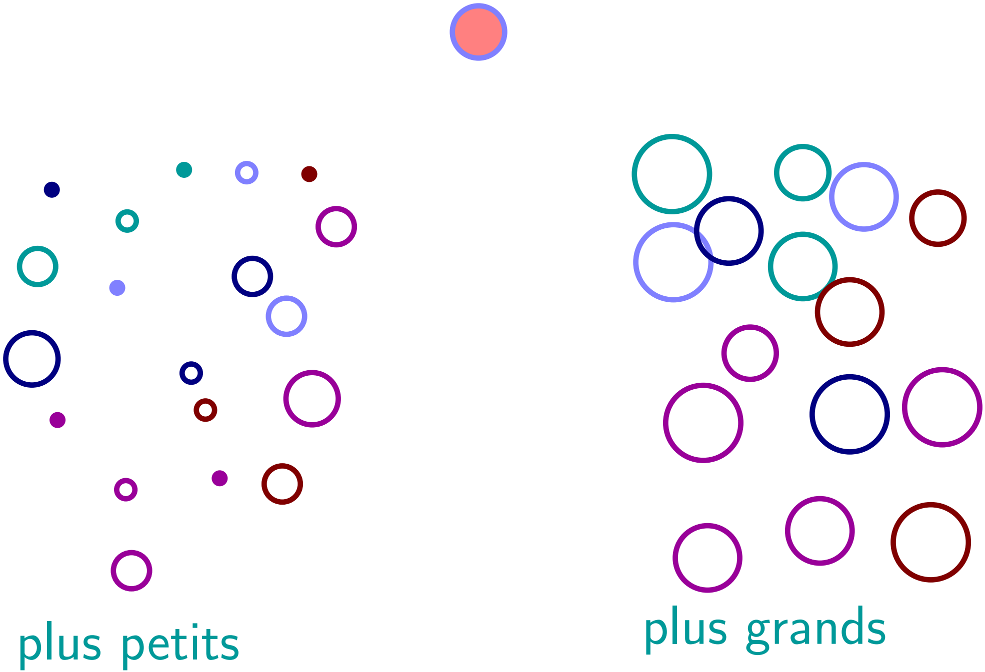
Médian linéaire

Médian linéaire

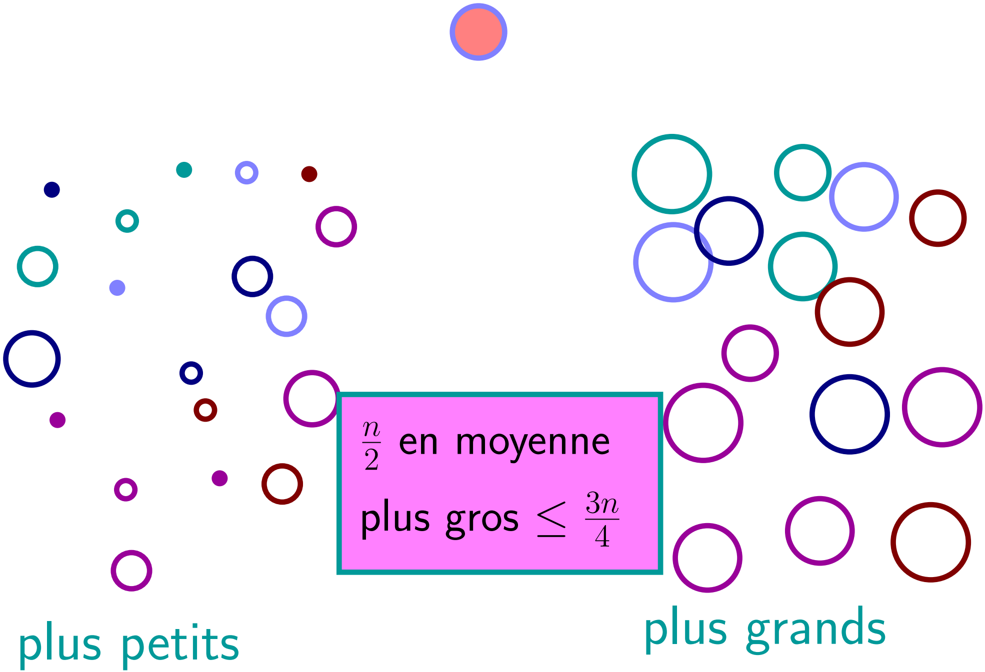


plus petits

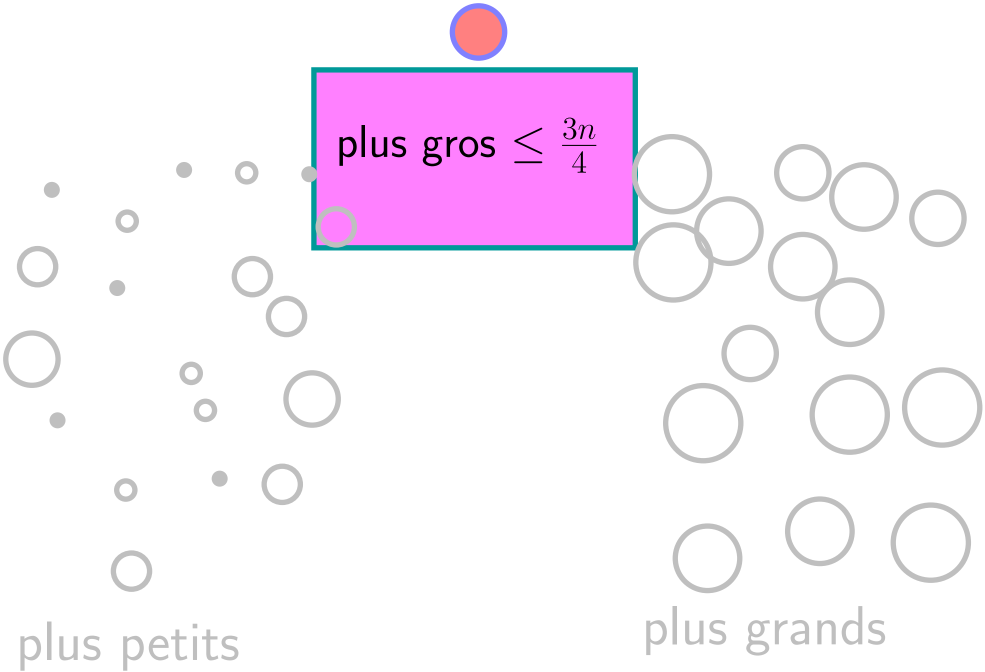
Médian linéaire



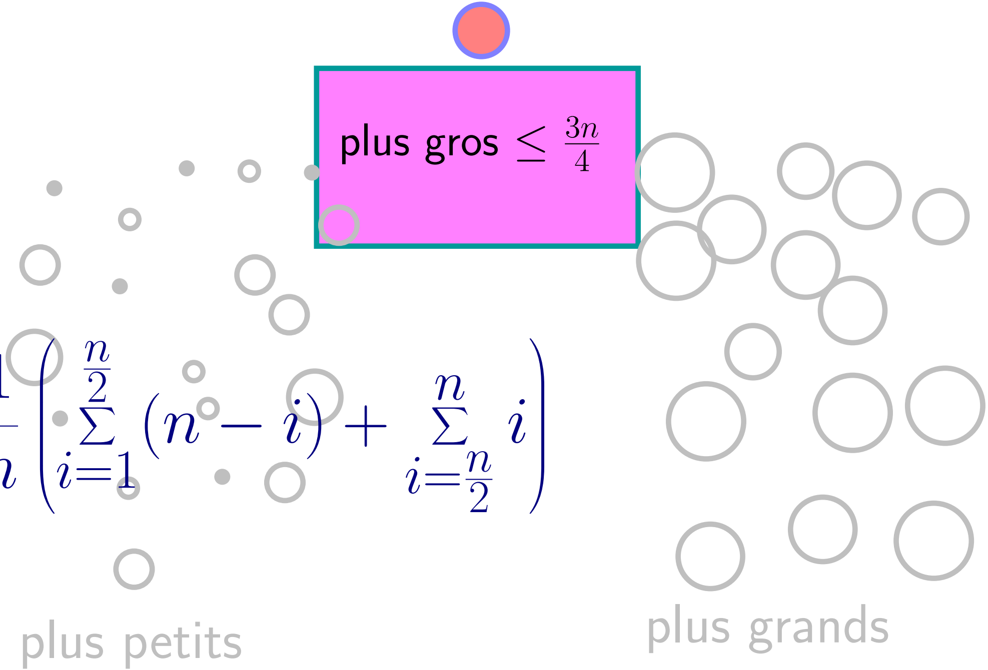
Médian linéaire



Médian linéaire



Médian linéaire



$$\frac{1}{n}\left(\sum_{i=1}^{\frac{n}{2}}(n-i)+\sum_{i=\frac{n}{2}}^{n} i\right)$$

Recherche du $k$ième élément

si rang( ) $> k$ on cherche le $k$ième élément

si rang( ) $= k$ on a trouvé

si rang( ) $< k$ on cherche le $(k-r)$ième

récursivement

Complexité

$$f(n) \leq n + f\left(\frac{3n}{4}\right)$$

Complexité

$$f(n) \leq n + f\left(\frac{3n}{4}\right)$$

$$f(n) \leq n + \frac{3n}{4} + f\left(\frac{9n}{16}\right)$$

Complexité

$$f(n) \leq n + f\left(\frac{3n}{4}\right)$$

$$f(n) \leq n + \frac{3n}{4} + f\left(\frac{9n}{16}\right)$$

$$f(n) \leq n \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i = 4$$

Complexité

$$f(n) \leq n + f\left(\frac{3n}{4}\right)$$

$$f(n) \leq n + \frac{3n}{4} + f\left(\frac{9n}{16}\right)$$

$$f(n) \leq n \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^{i} = 4$$

Médian linéaire randomisé

Complexité

$$f(n) \leq n + f\left(\frac{3n}{4}\right)$$

$$f(n) \leq n + \frac{3n}{4} + f\left(\frac{9n}{16}\right)$$

$$f(n) \leq n \sum_{i=0}^{\infty} \left(\frac{3}{4}\right)^i = 4$$

Médian linéaire randomisé

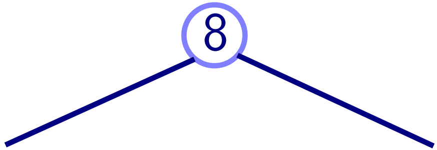
Dans le cas le pire

Quadratique

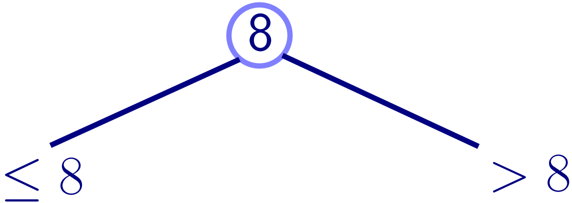
Arbre binaire de recherche

8

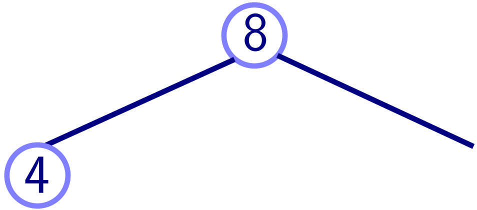
Arbre binaire de recherche



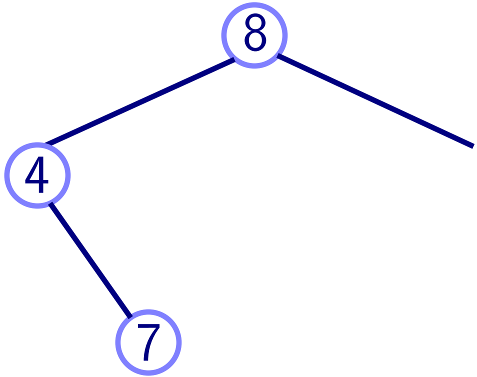
Arbre binaire de recherche



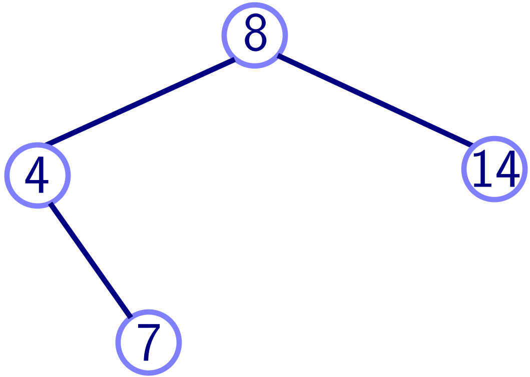
Arbre binaire de recherche



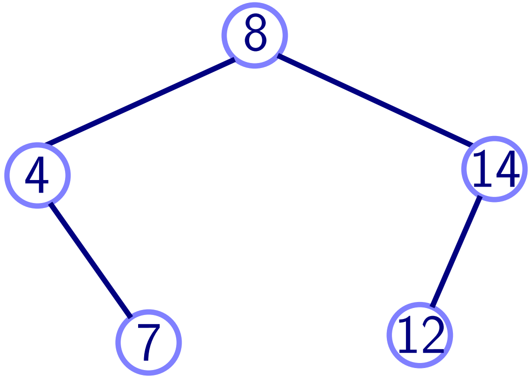
Arbre binaire de recherche



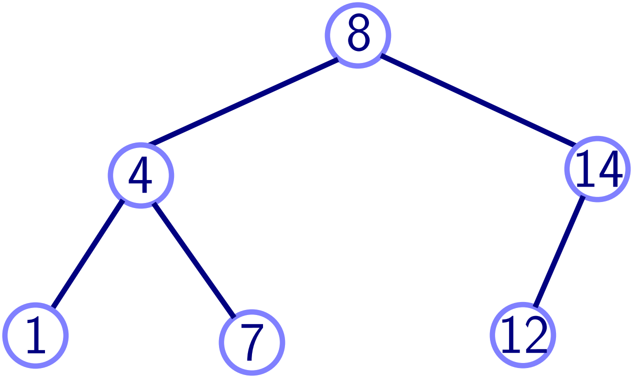
Arbre binaire de recherche



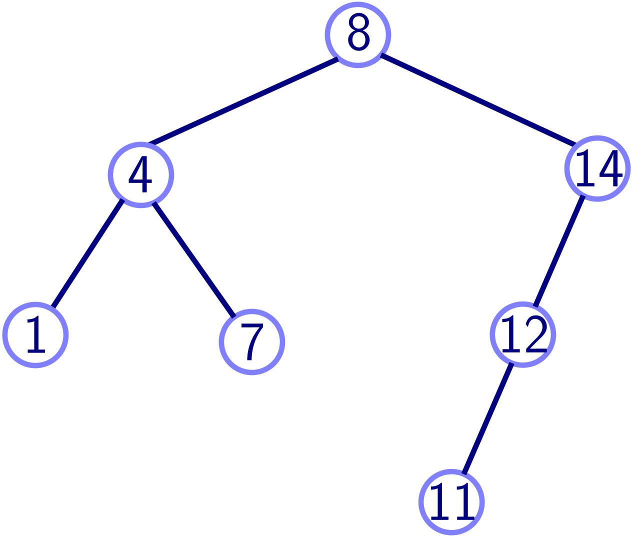
Arbre binaire de recherche



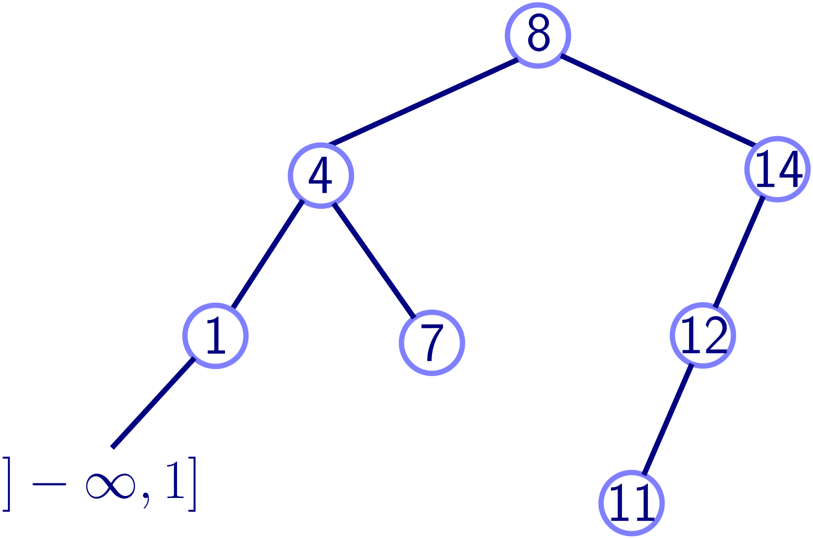
Arbre binaire de recherche



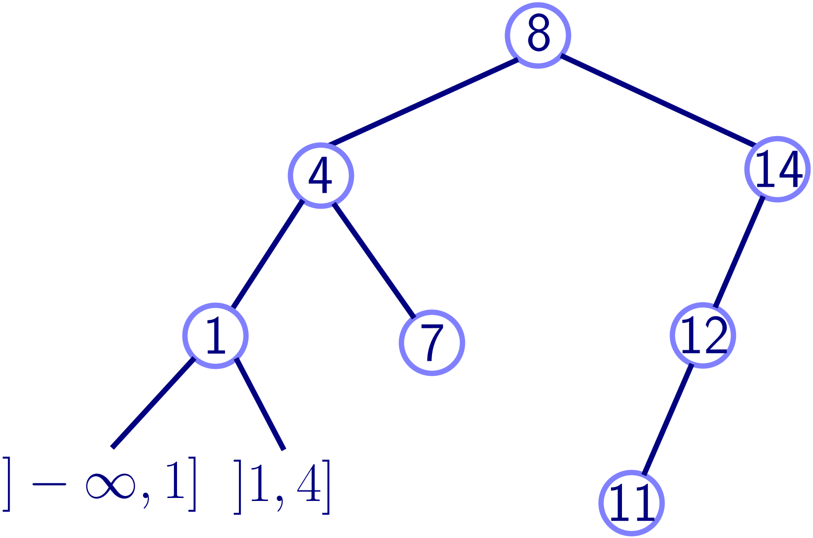
Arbre binaire de recherche



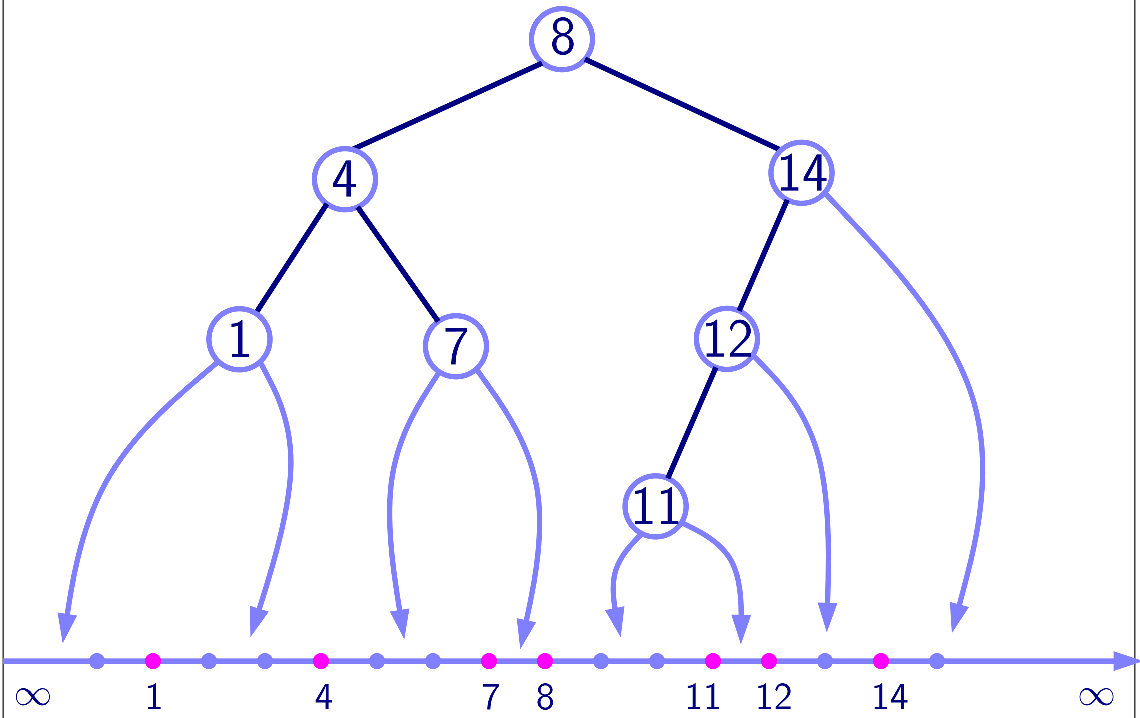
Arbre binaire de recherche



Arbre binaire de recherche



Arbre binaire de recherche

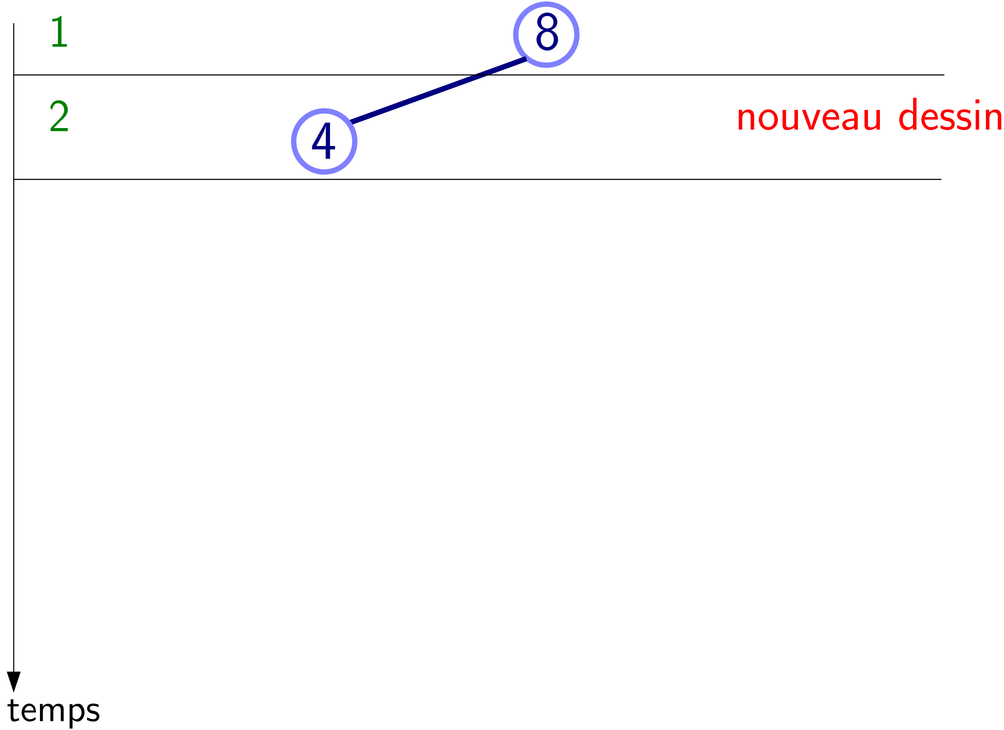


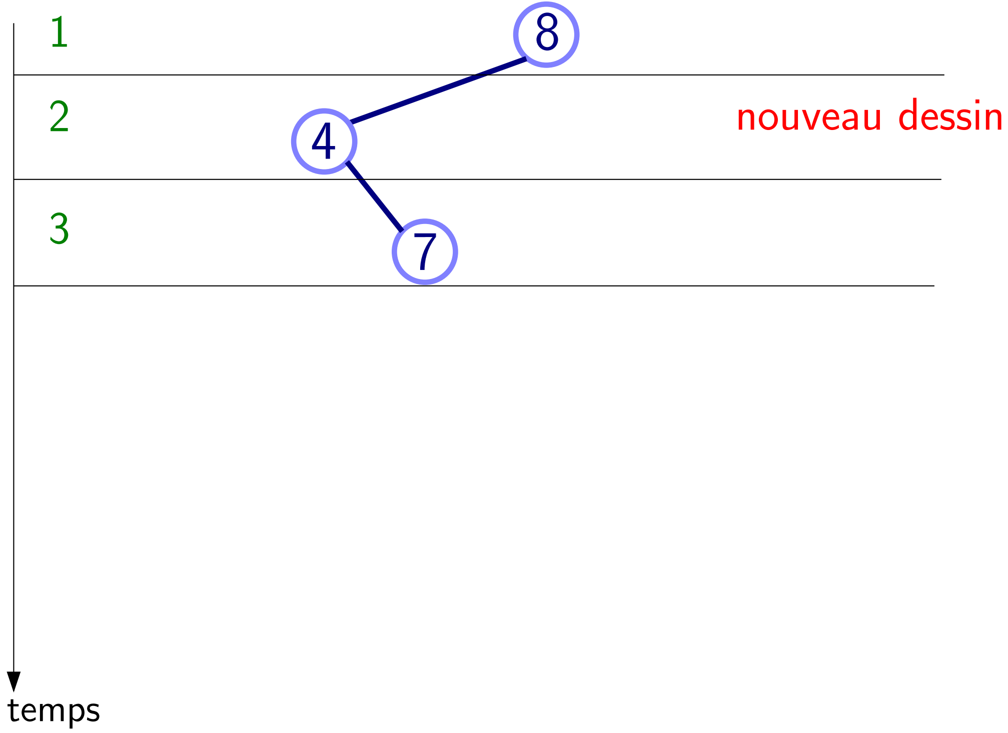
1

8

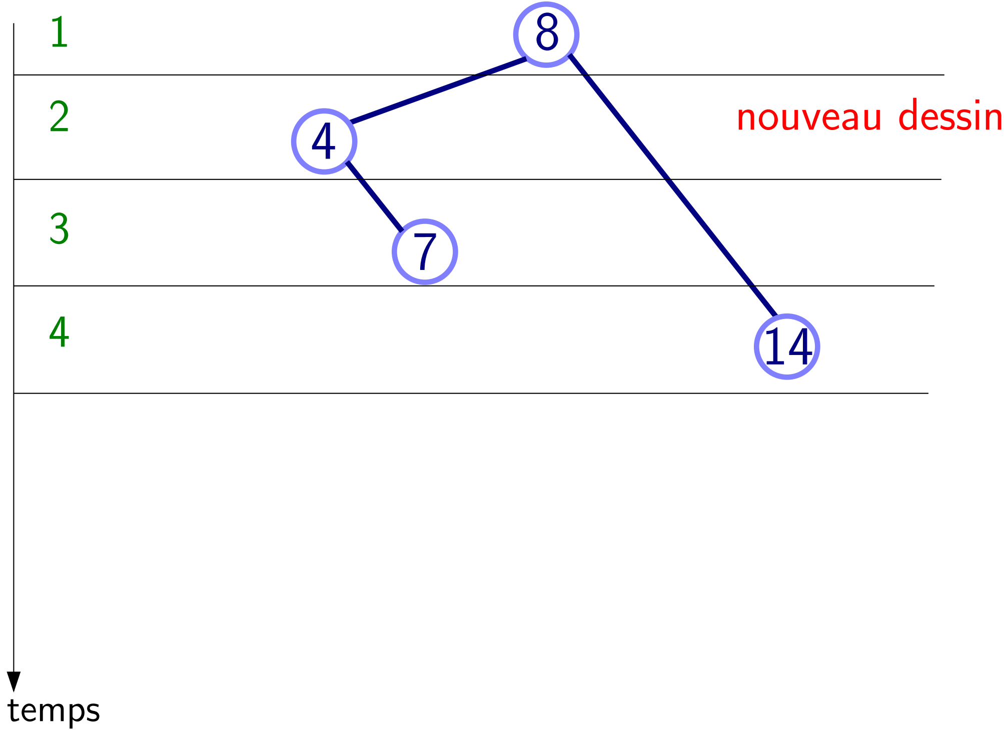
nouveau dessin

temps

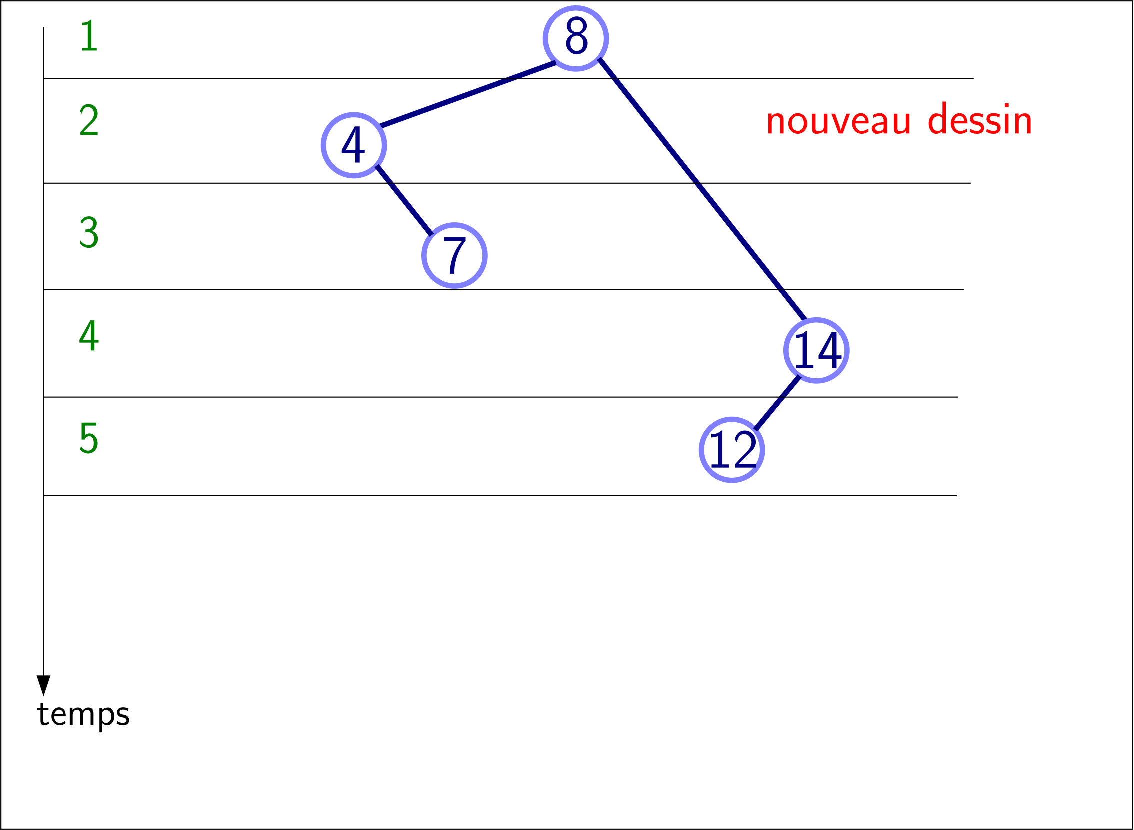

1
8
2
4
nouveau dessin
temps

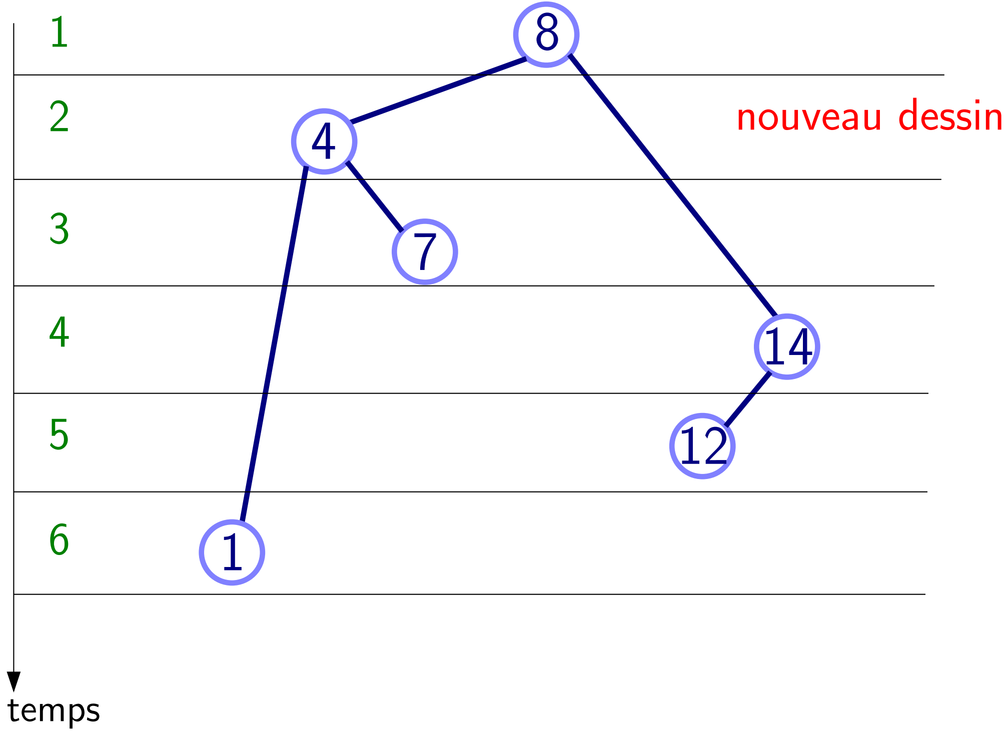


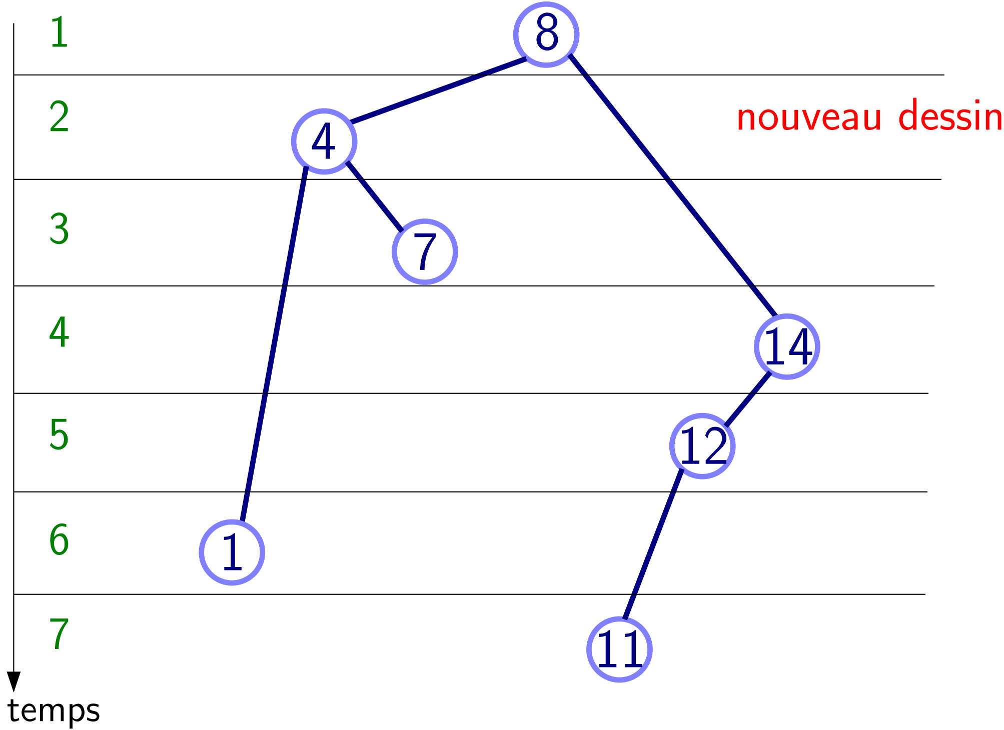
1
8
2
4
nouveau dessin
3
7
temps

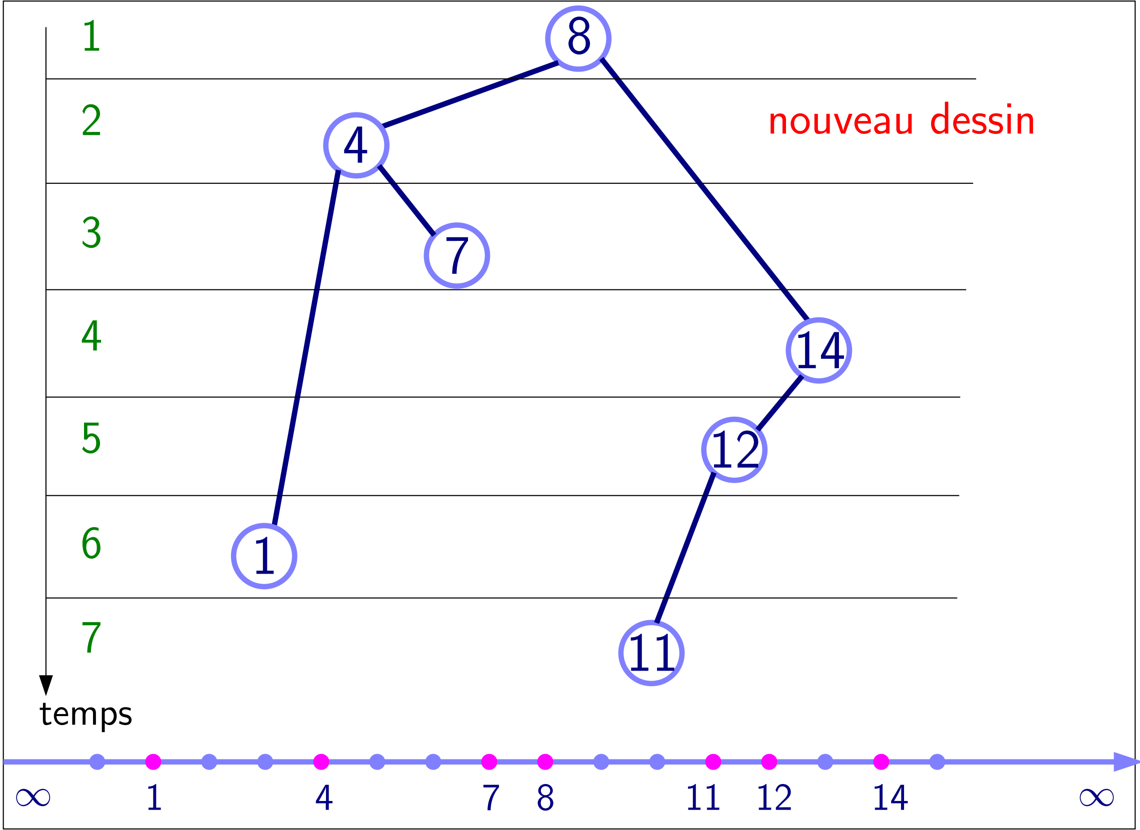


1
2
3
4
8
4
7
14
nouveau dessin
temps

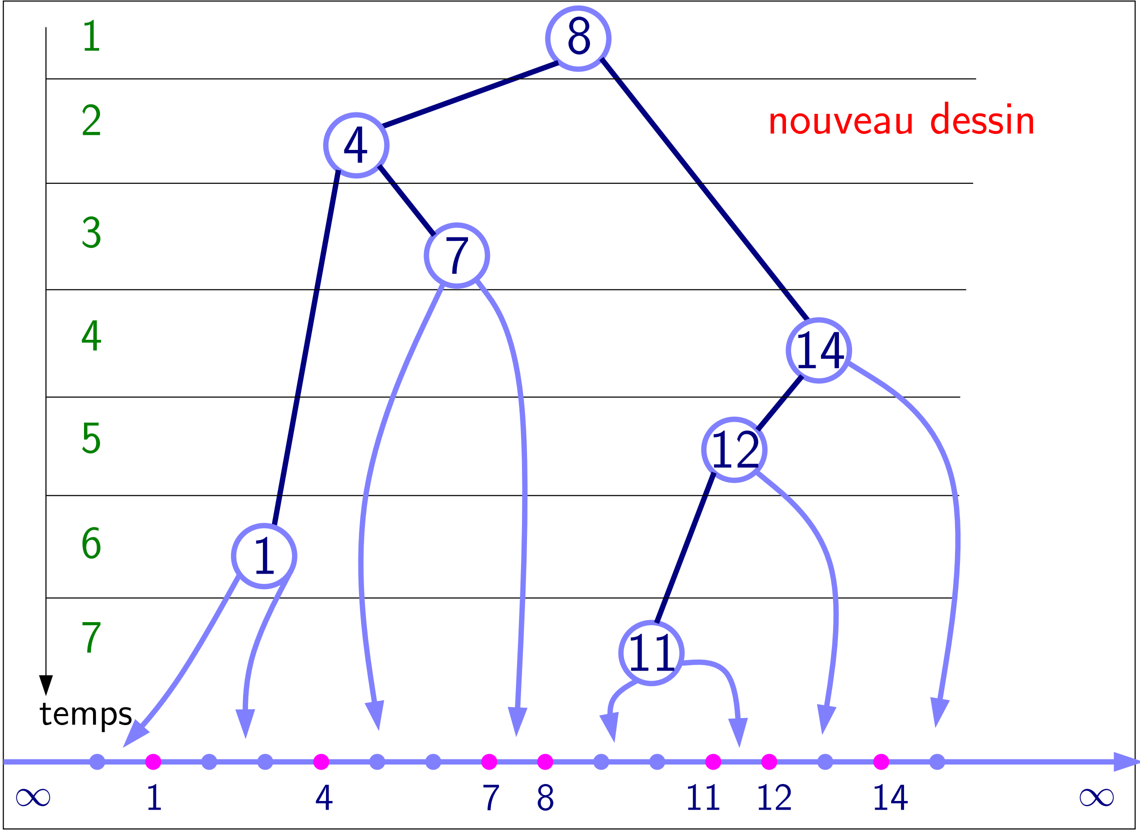

1
2
3
4
5
temps
8
4
7
14
12
nouveau dessin

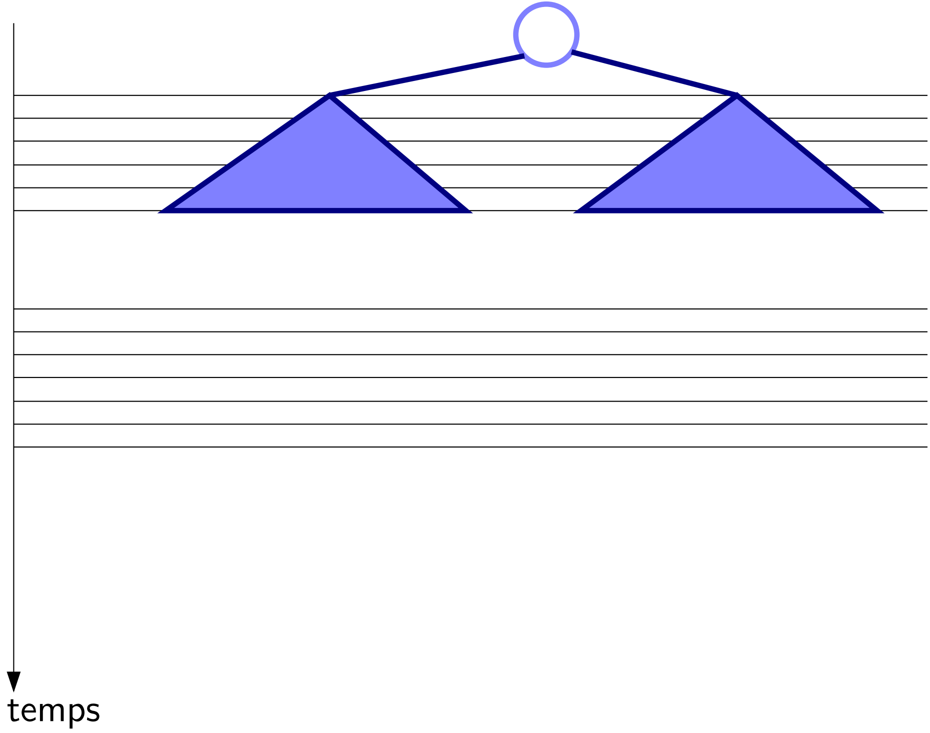

1
2
3
4
5
6
8
4
7
14
12
1
nouveau dessin
temps

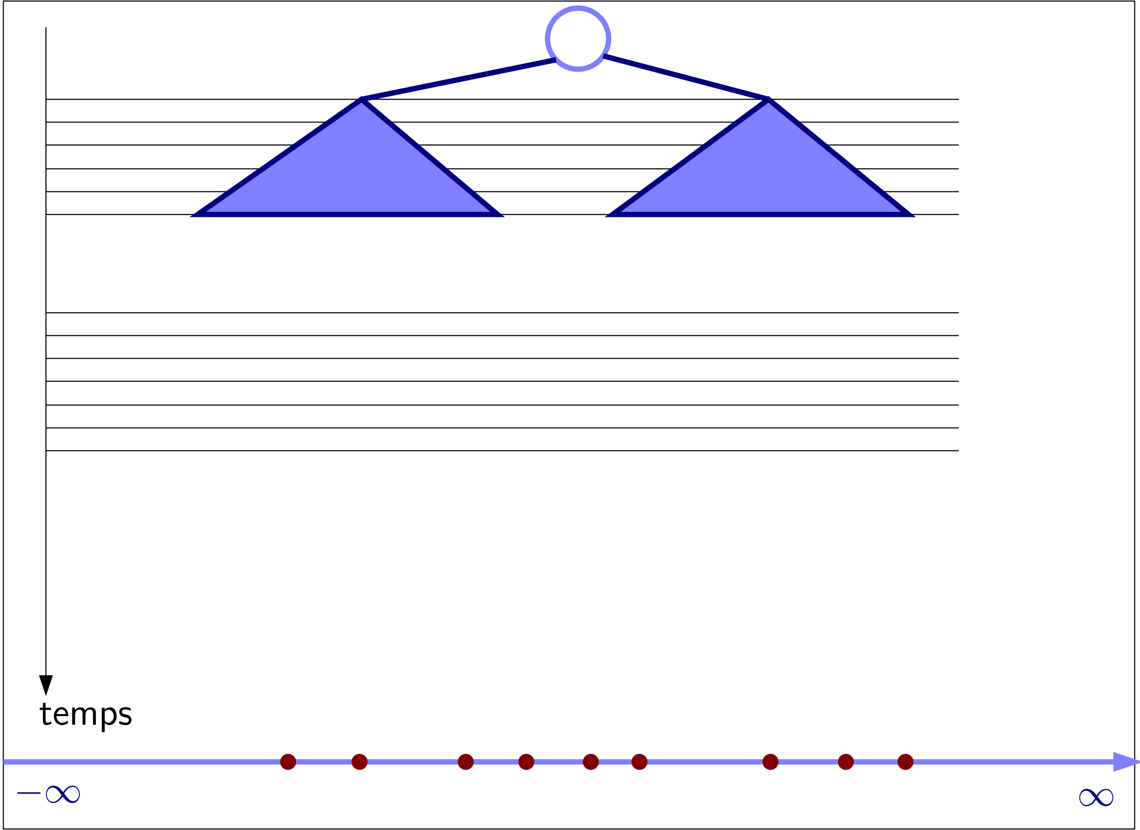

1
2
3
4
5
6
7
temps
nouveau dessin
8
4
7
14
12
1
11



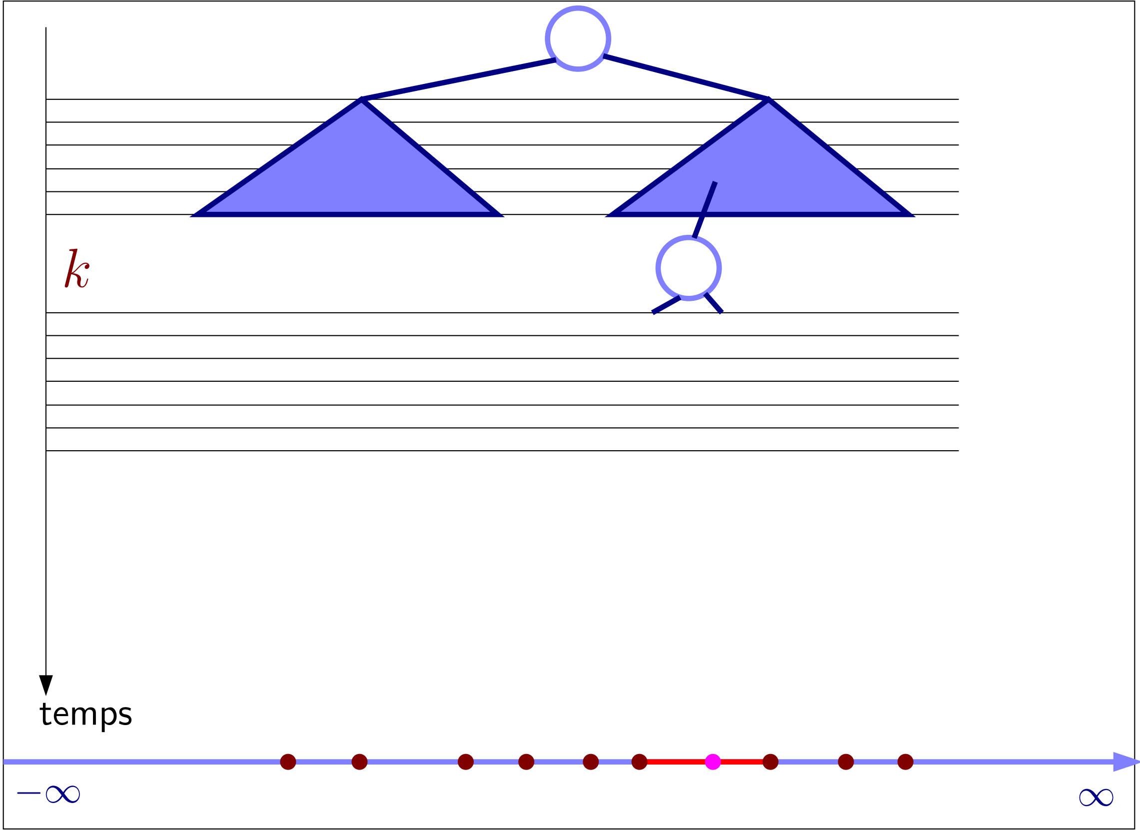
1
2
3
4
5
6
7
temps
nouveau dessin
8
4
7
14
12
1
11
∞
1
4
7
8
11
12
14
∞


1
2
3
4
5
6
7
temps
nouveau dessin
8
4
7
14
12
1
11
∞
1
4
7
8
11
12
14
∞

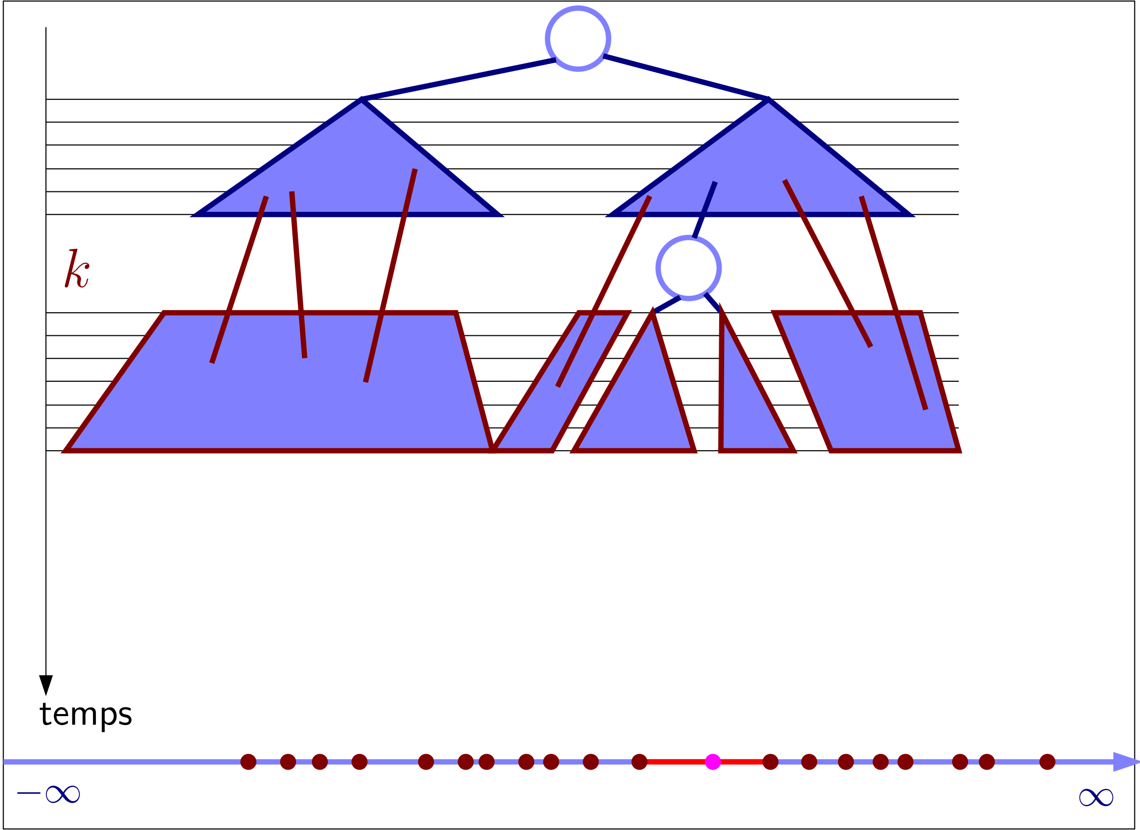

temps



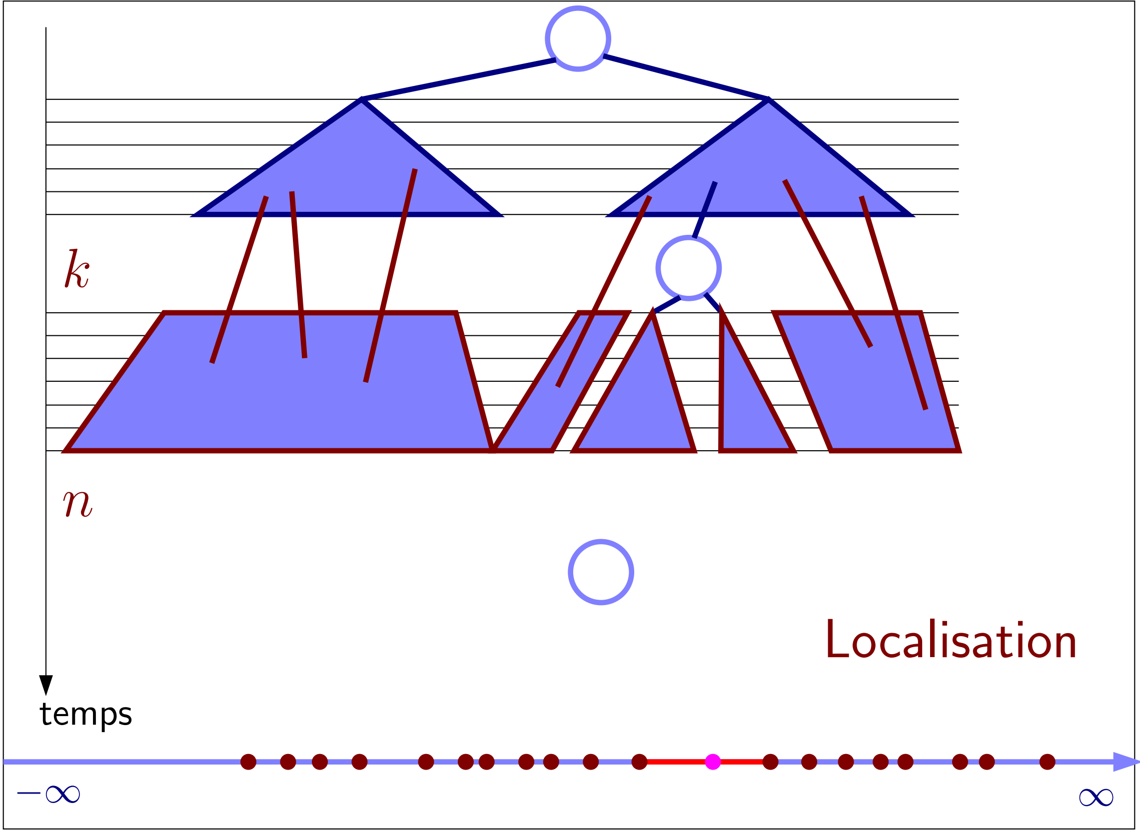
temps
$-\infty$
∞



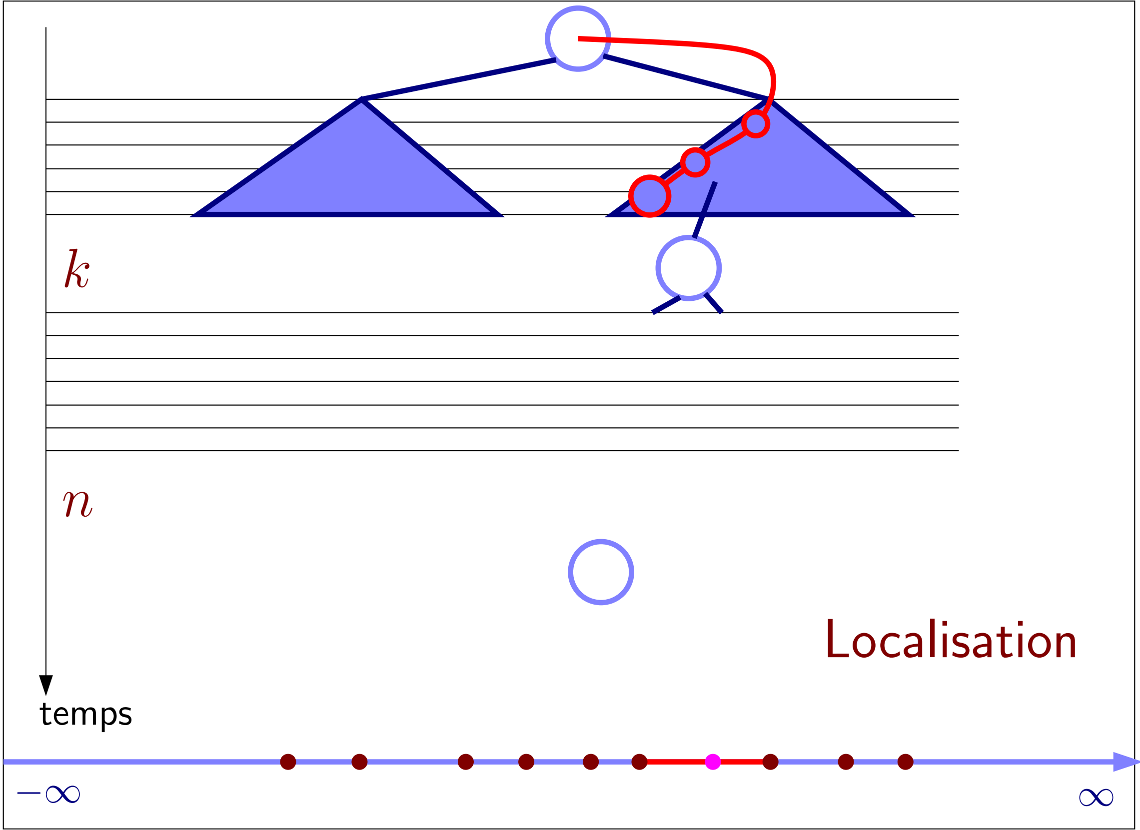
k
temps
$-\infty$
∞

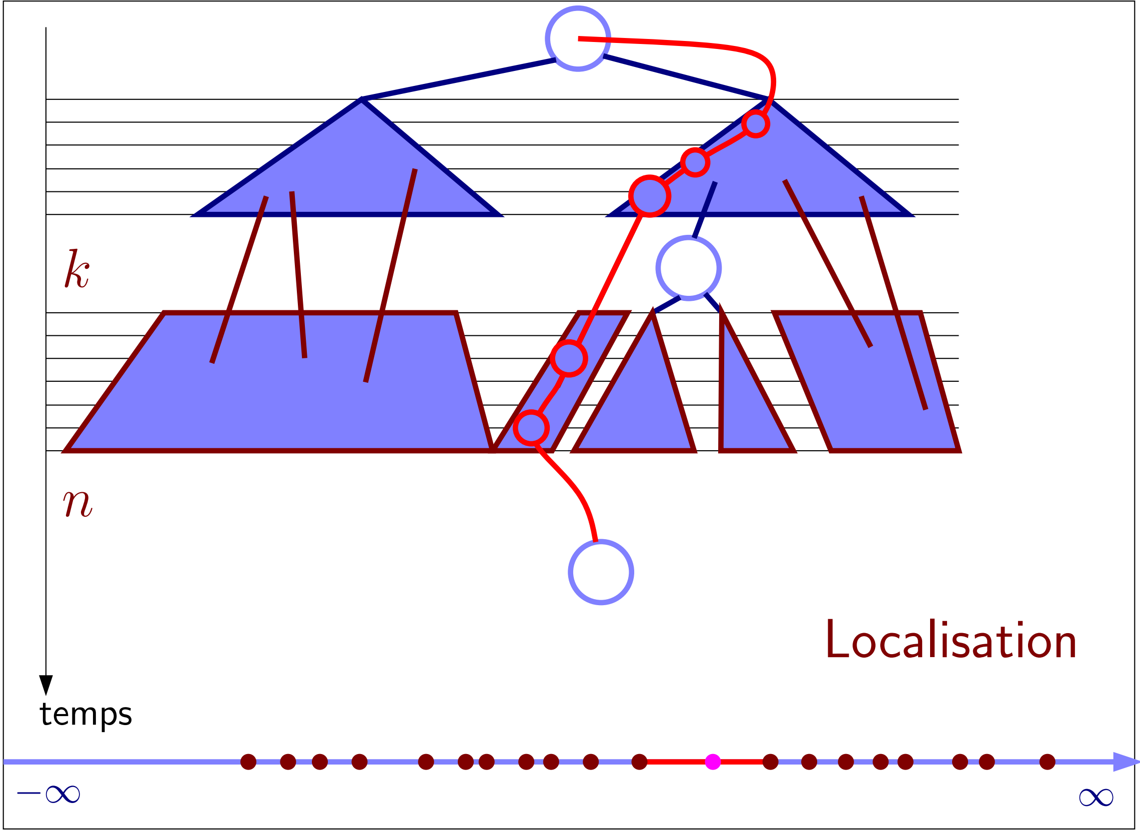


k
temps
$-\infty$
∞

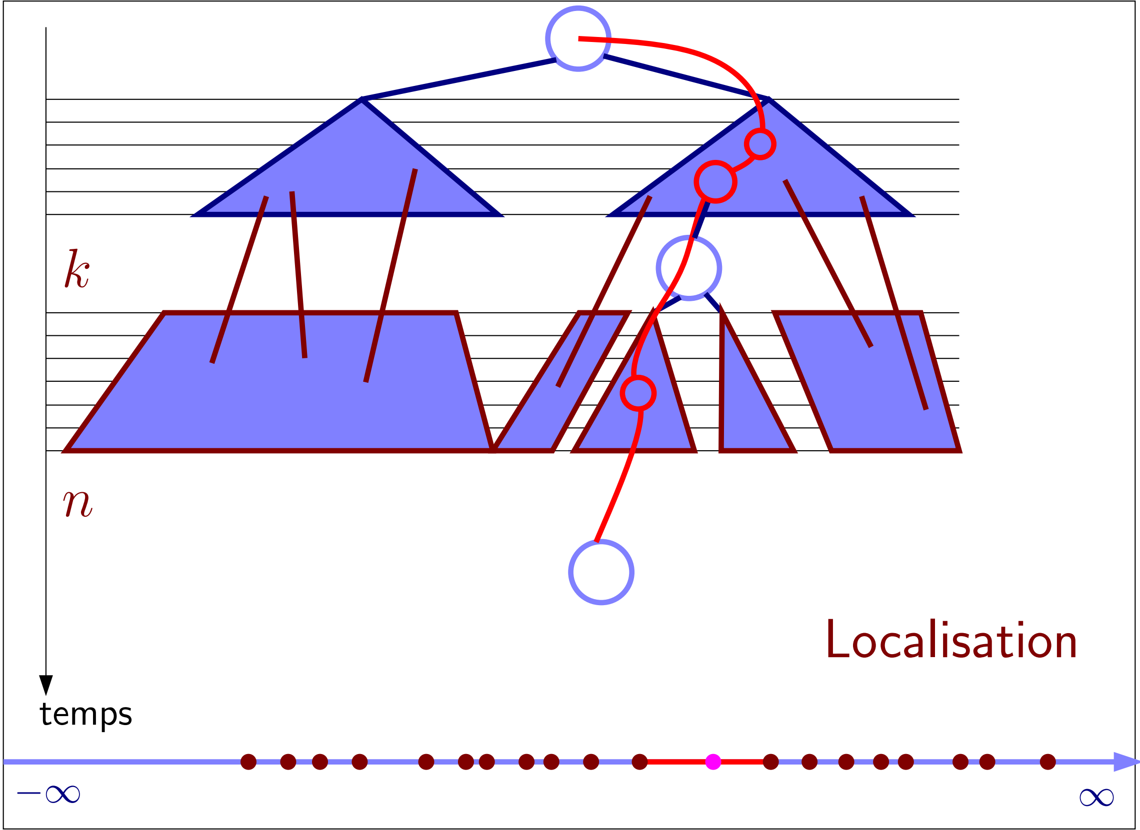


k
n
Localisation
temps
$-\infty$
∞

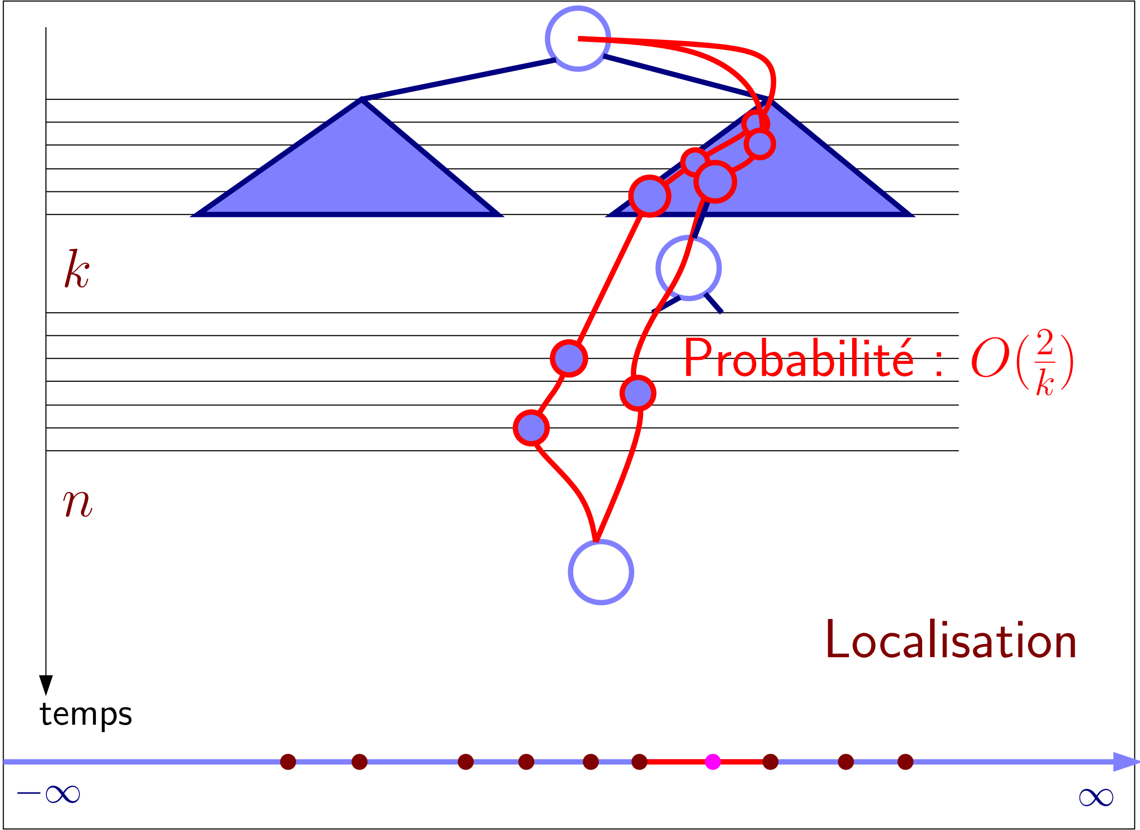

k
n
Localisation
temps
$-\infty$
∞



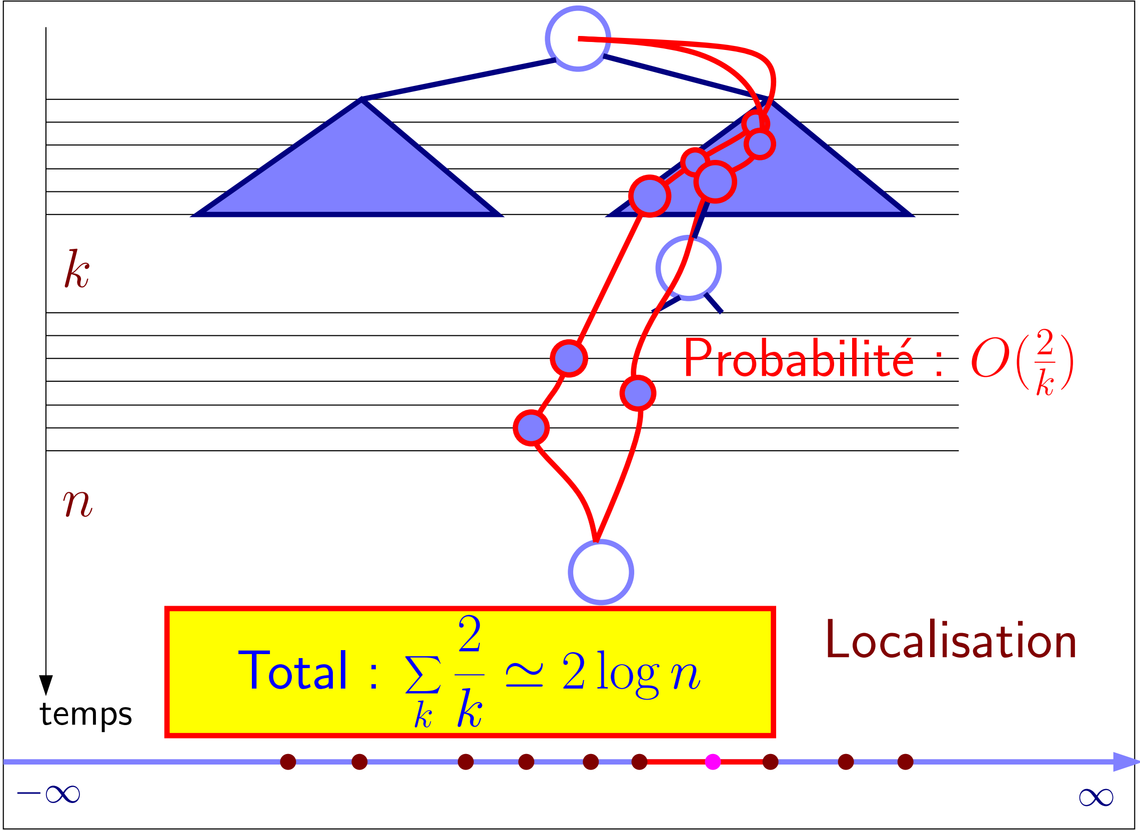
k
n
Localisation
temps
$-\infty$
∞



k
n
Localisation
temps
$-\infty$
∞



k
n
Probabilité : $O(\frac{2}{k})$
Localisation
temps
$-\infty$
∞



k
n
temps
Probabilité : $O(\frac{2}{k})$
Total : $\sum_k \frac{2}{k} \simeq 2\log n$
Localisation
$-\infty$
∞

Arbre binaire (non équilibré)

Insertion dans un ordre aléatoire

Arbre binaire (non équilibré)

Insertion dans un ordre aléatoire

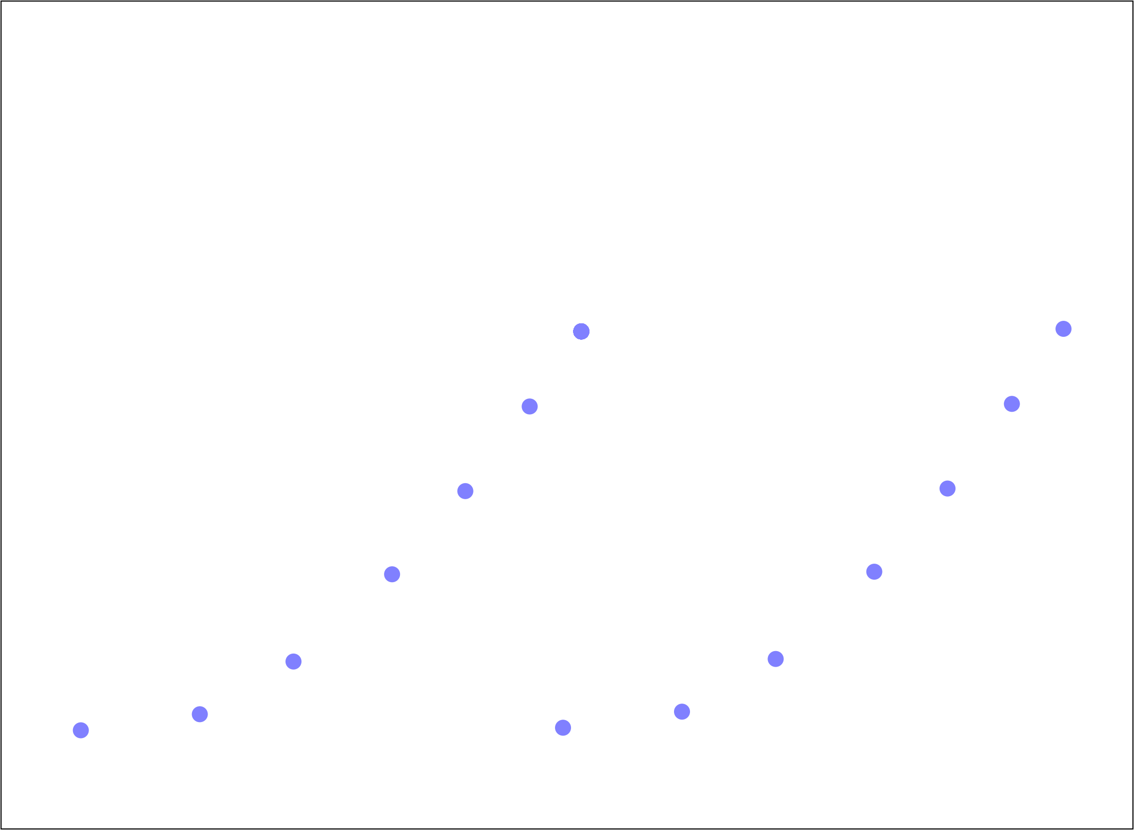
$O(n \log n)$

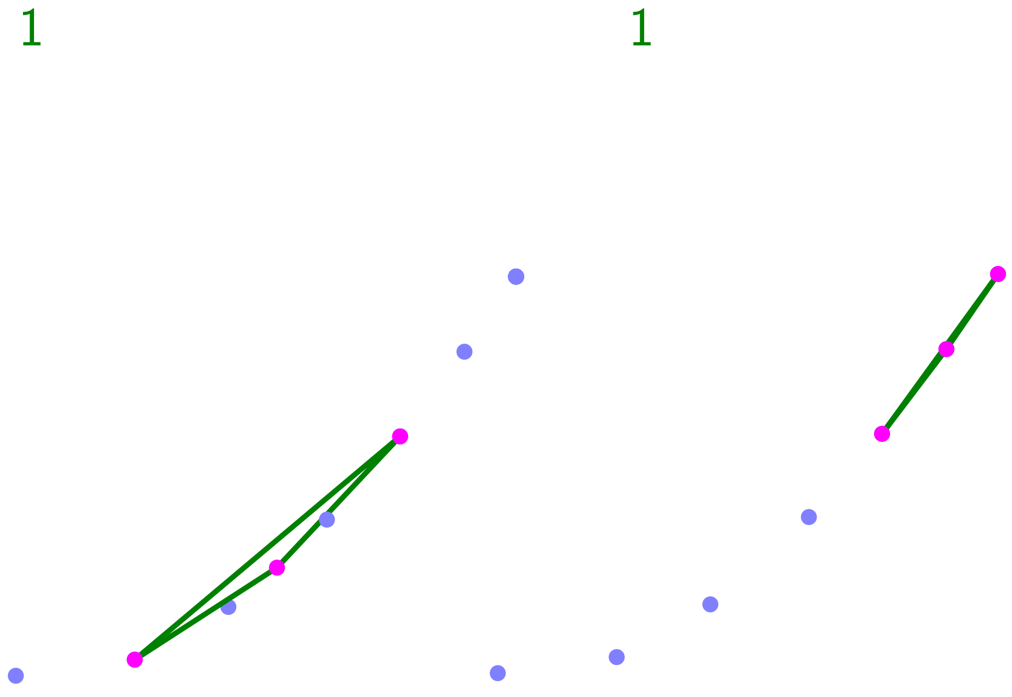
s'applique à quicksort

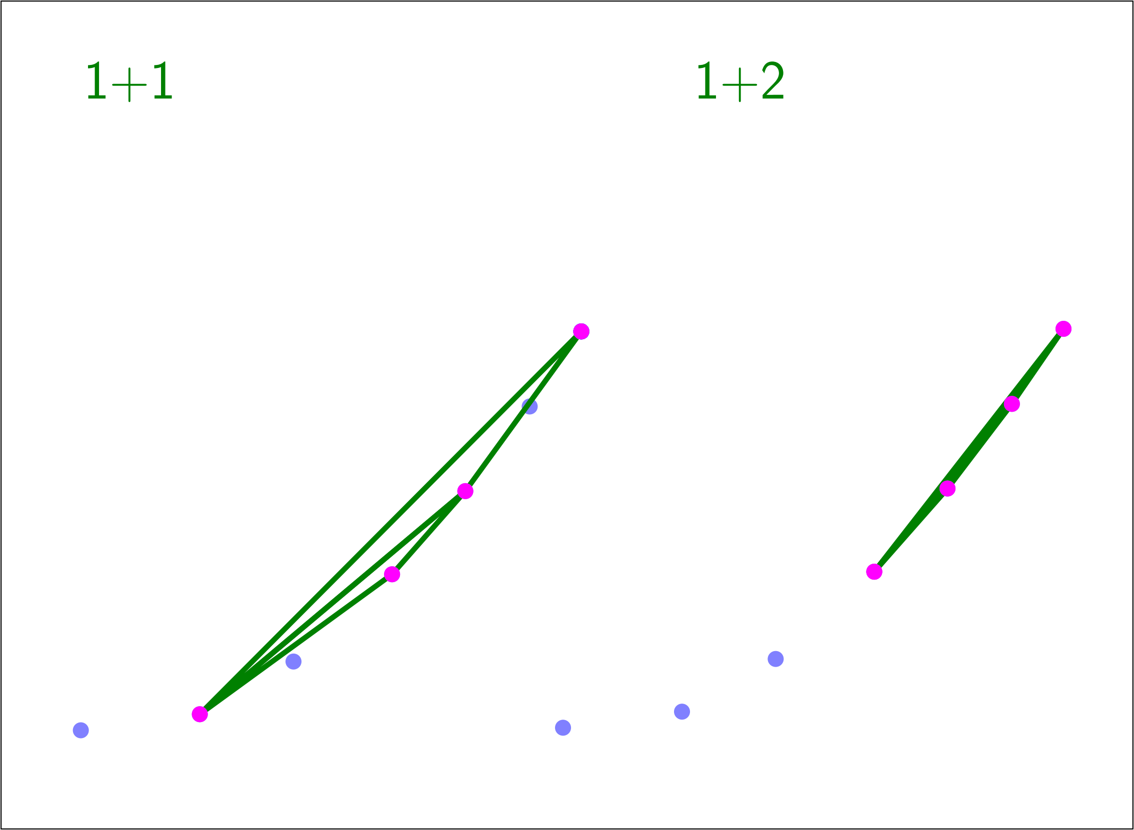
Borne inférieure pour la triangulation de Delaunay incrémentale

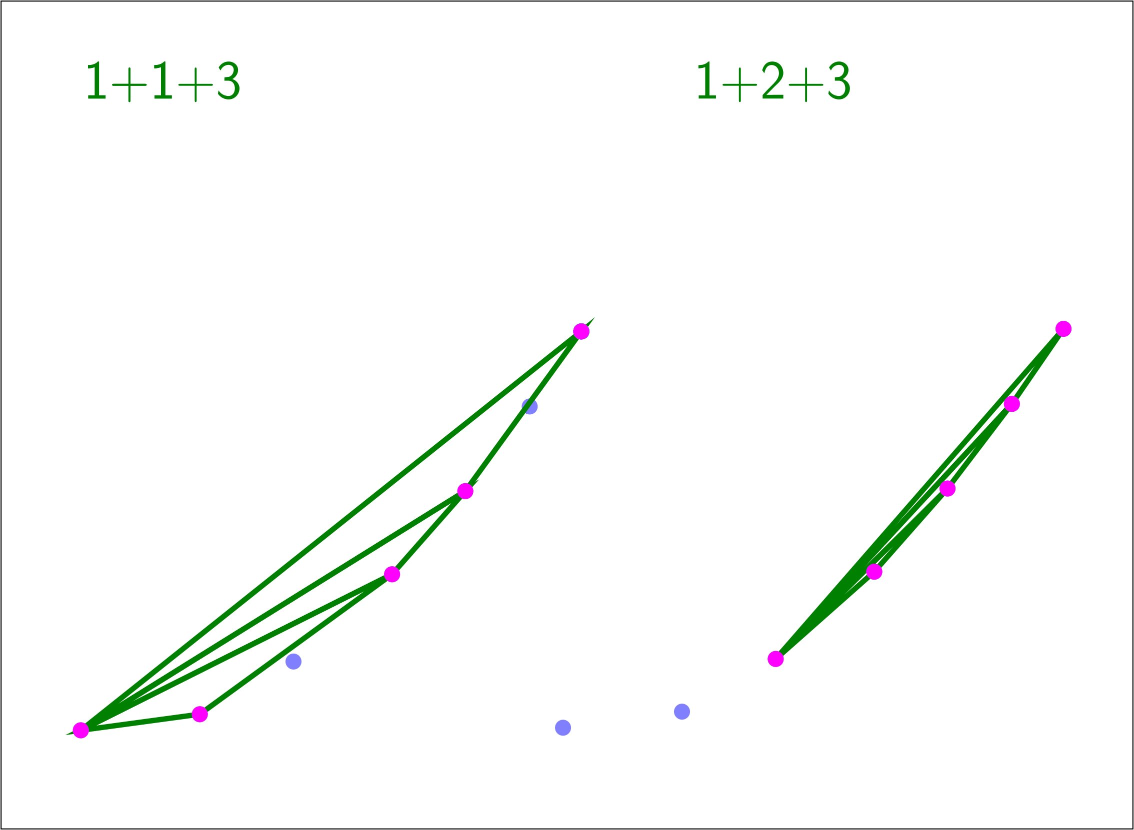
nombre de triangles crés

quadratique



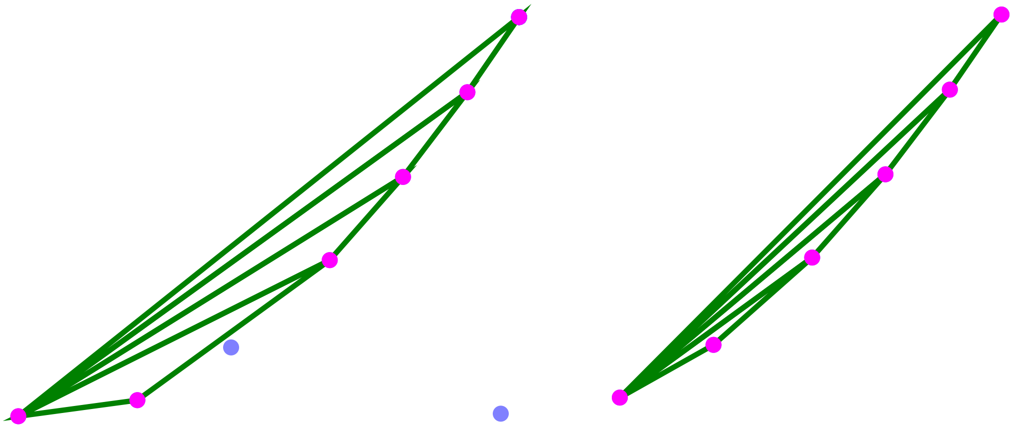

1
1


1+1
1+2

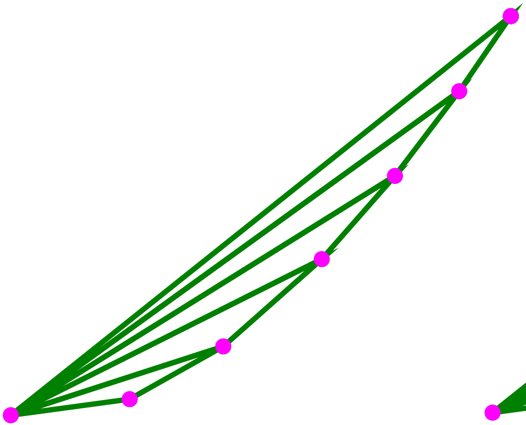


1+1+3
1+2+3

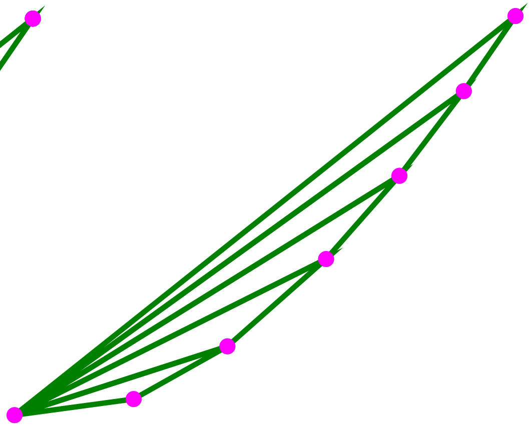
$1+1+3+2$ $\qquad$ $1+2+3+4$



$$1+1+3+2+2=9 \qquad\qquad 1+2+3+4+5=15$$



????



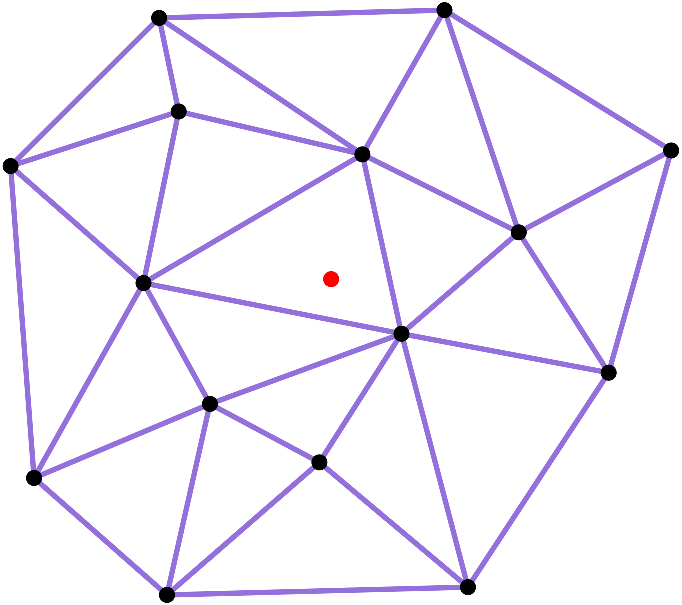
quadratique

Analyse randomisée pour

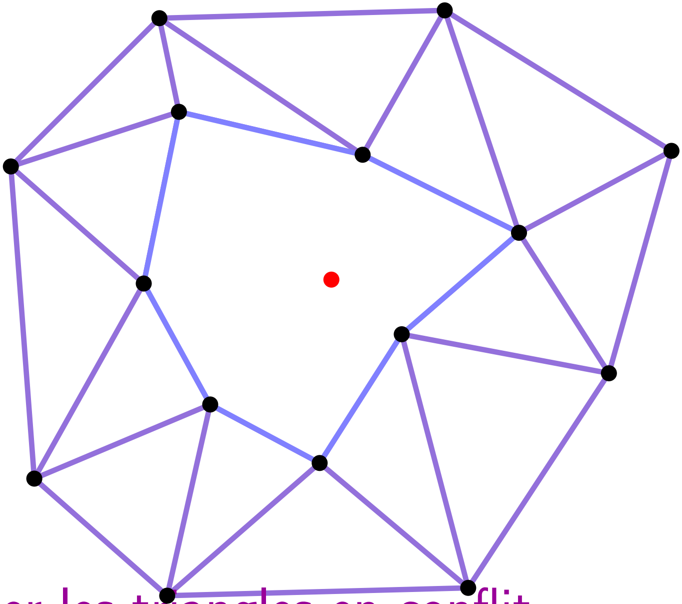
la triangulation de Delaunay

l'insertion

Algorithme incrémental

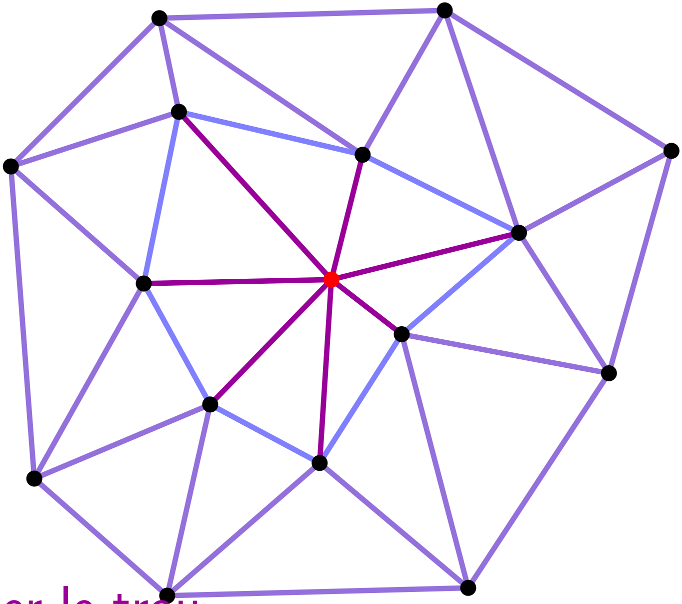


Algorithme incrémental



Supprimer les triangles en conflit

Algorithme incrémental



Trianguler le trou

Insertion des points dans un ordre aléatoire

Coût du dernier point ?

Insertion des points dans un ordre aléatoire

Coût du dernier point ?

Degré du dernier point ?

Insertion des points dans un ordre aléatoire

Coût du dernier point ?

Degré du dernier point ?

Degré d'un point aléatoire ?

Insertion des points dans un ordre aléatoire

Coût du dernier point ?

Degré du dernier point ?

Degré d'un point aléatoire ?

$$\frac{1}{n}\sum_{i=1}^{n} \text{degré}(p_i)$$

Insertion des points dans un ordre aléatoire

Coût du dernier point ?

Degré du dernier point ?

Degré d'un point aléatoire ?

$$\frac{1}{n}\sum_{i=1}^{n} \text{degré}(p_i) \quad \leq 6$$

Euler

Insertion des points dans un ordre aléatoire

Coût du dernier point ≤ 6

Insertion des points dans un ordre aléatoire

Coût du dernier point ≤ 6

Coût total

Insertion des points dans un ordre aléatoire

Coût du dernier point ≤ 6

Coût total $\quad \leq \sum_{1}^{n} 6 = O(n)$

Analyse randomisée pour

la triangulation de Delaunay

la localisation

Arbre de Delaunay

Ce qui est cher

c'est la localisation

Structures de données

pour la localisation

Arbre de Delaunay

(pas le mieux)

(raisonablement simple)

Se souvenir de l'historique

de la construction de

la triangulation de Delaunay

Se souvenir de l'historique

de la construction de

la triangulation de Delaunay

Si $p \in Cercle(t)$

On dit t en conflit avec p

Se souvenir de l'historique

de la construction de

la triangulation de Delaunay

Si $p \in Cercle(t)$

On dit t en conflit avec p

Localiser $=$

Trouver les triangles en conflit

(les triangles de Delaunay)

Se souvenir de l'historique

de la construction de

la triangulation de Delaunay

Si $p \in Cercle(t)$

On dit t en conflit avec p

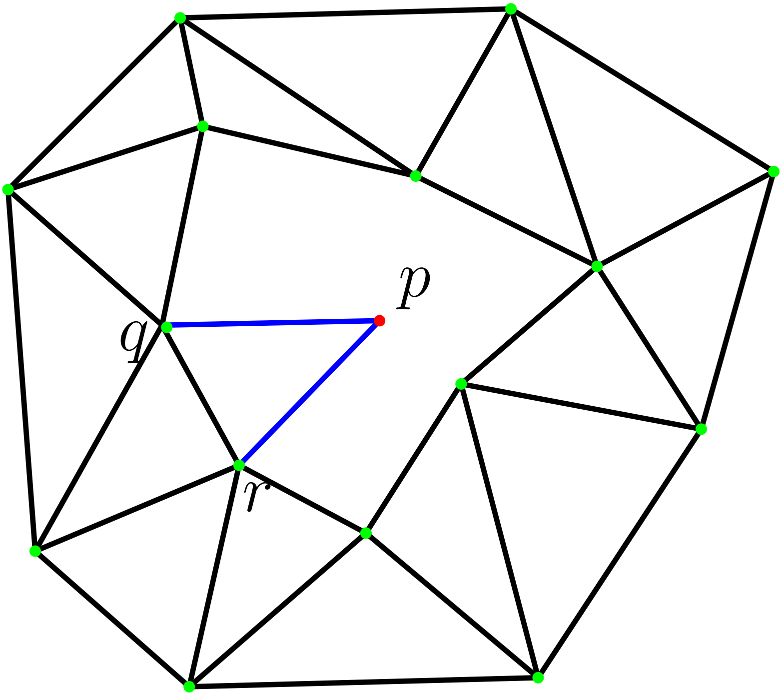
Localiser $=$

Trouver les triangles en conflit

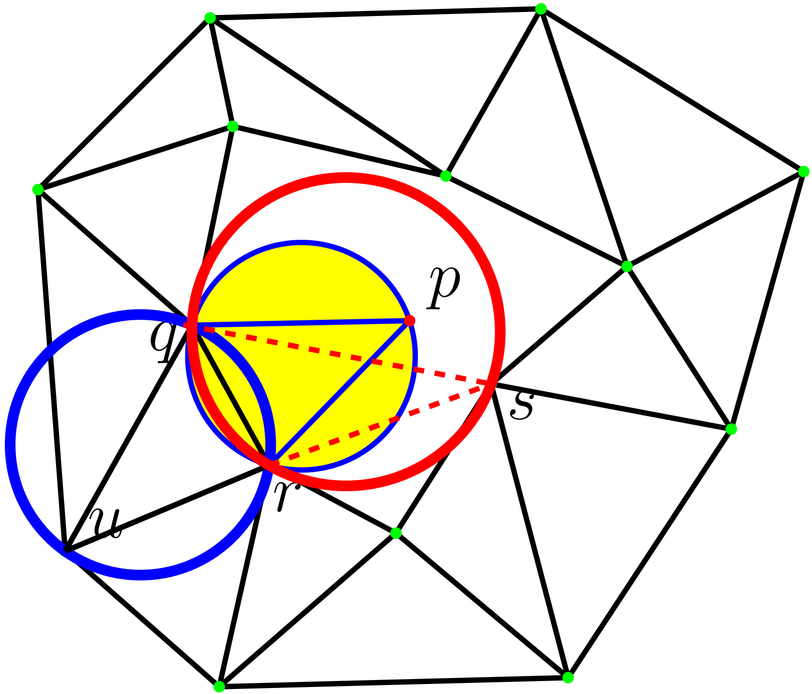
(les triangles de Delaunay)

triangles ayant été de Delaunay

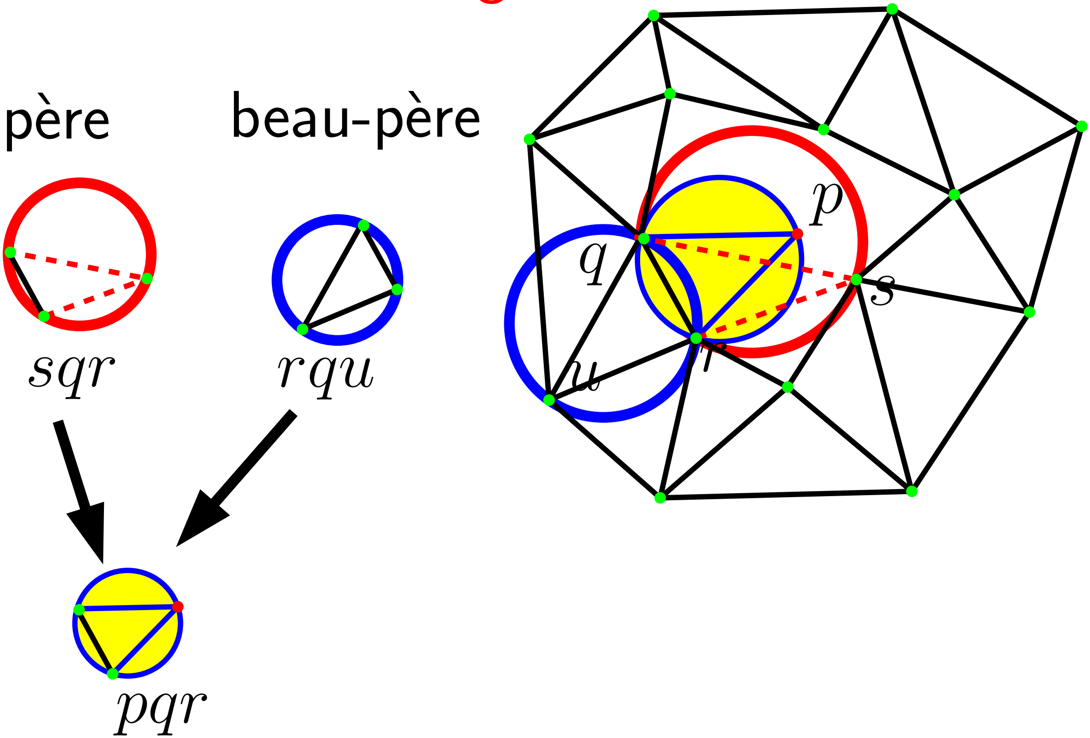
Création d'un triangle



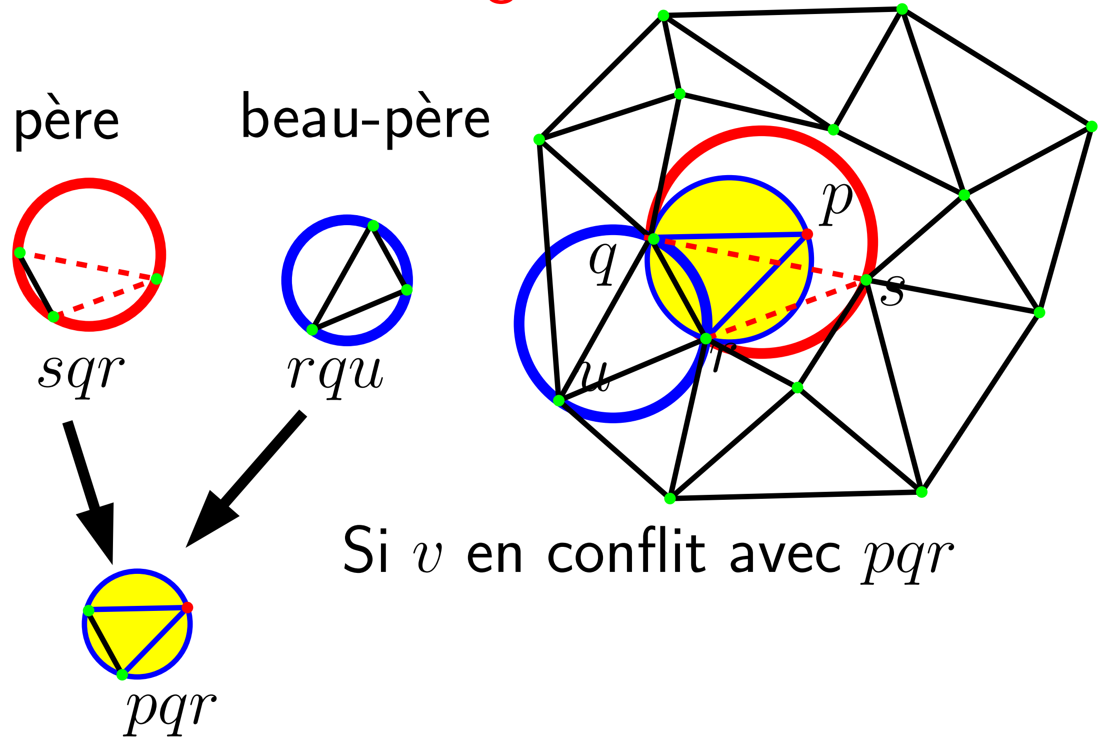
Création d'un triangle



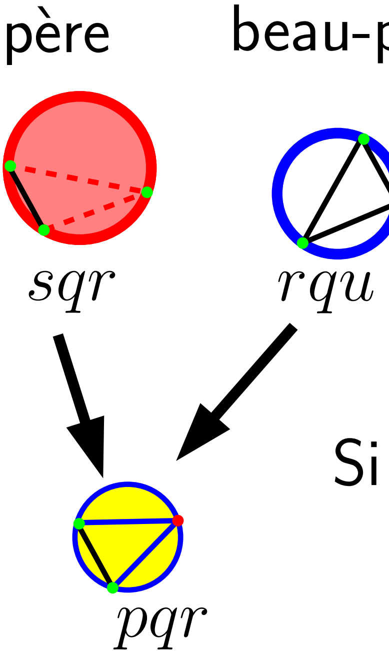
Création d'un triangle

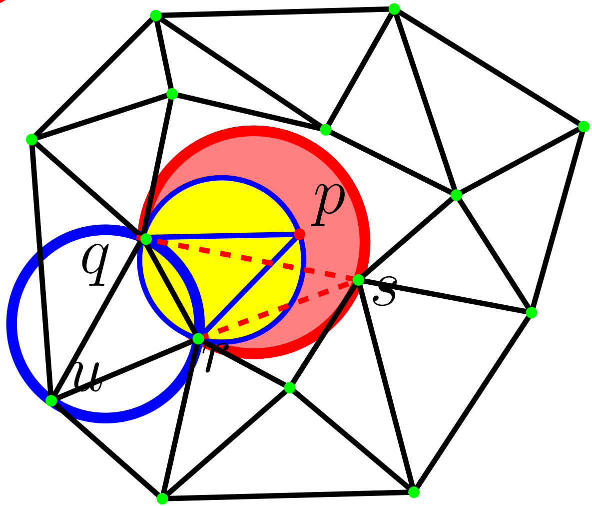


Création d'un triangle



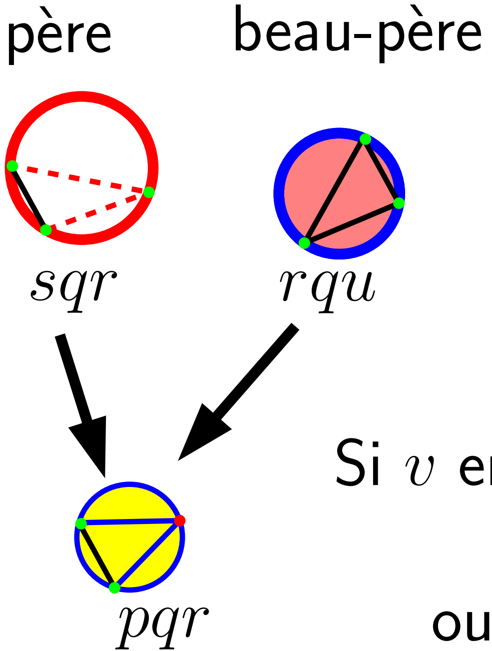
Création d'un triangle

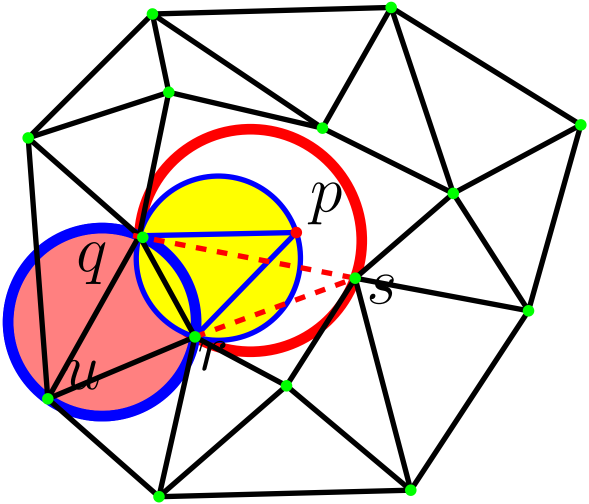




Si v en conflit avec pqr
v en conflit avec sqr

Création d'un triangle





Si v en conflit avec pqr

ou v en conflit avec rqu

Localisation de v

On trouve tous les triangles

ayant étés de Delaunay

en conflit avec v

Localisation de v

On trouve tous les triangles

ayant étés de Delaunay

en conflit avec v

Problème :

les compter !

(pour le dernier point)

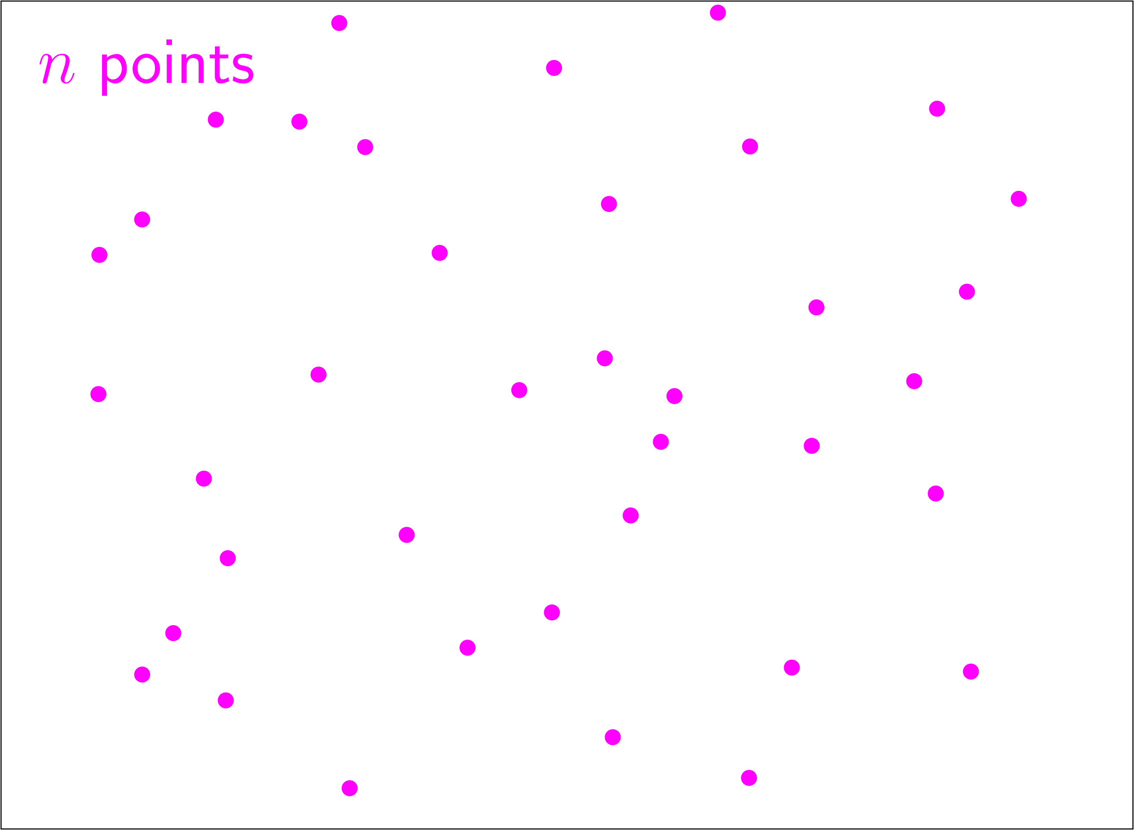
t un triangle ayant été Delaunay

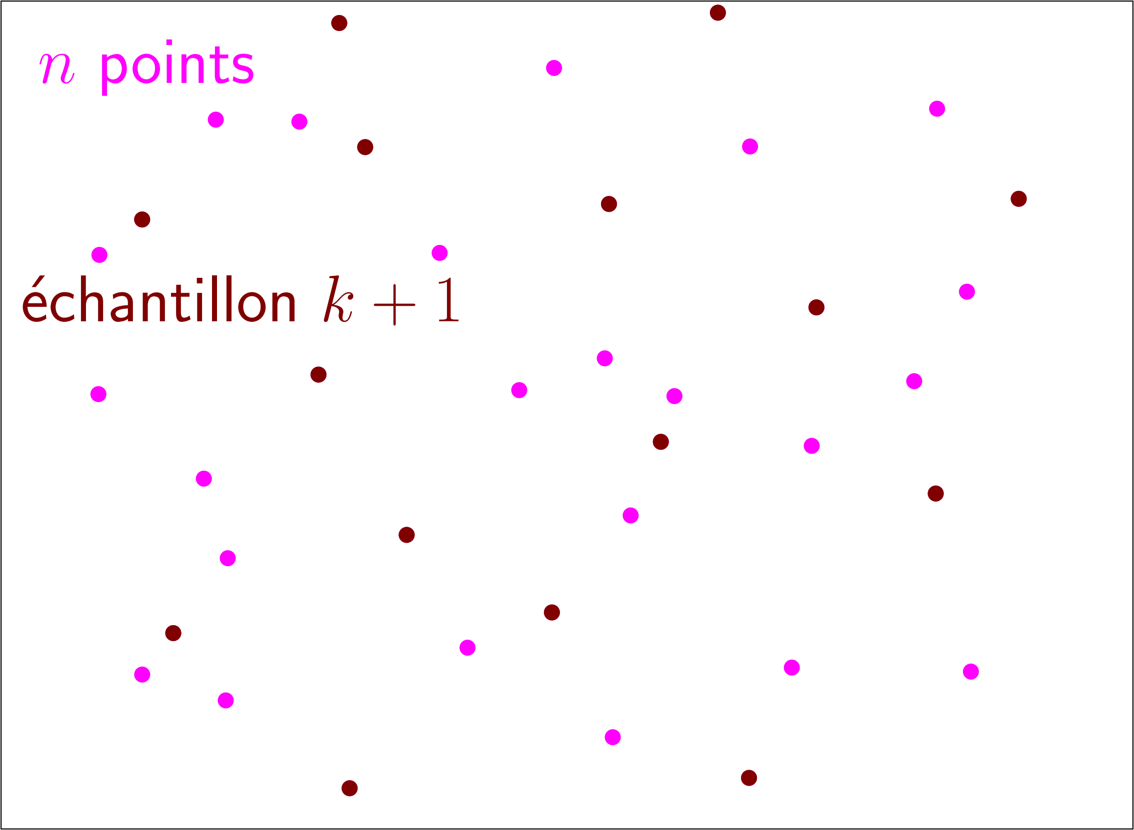
x_n le dernier point

Proba(x_n en conflit avec t) ?

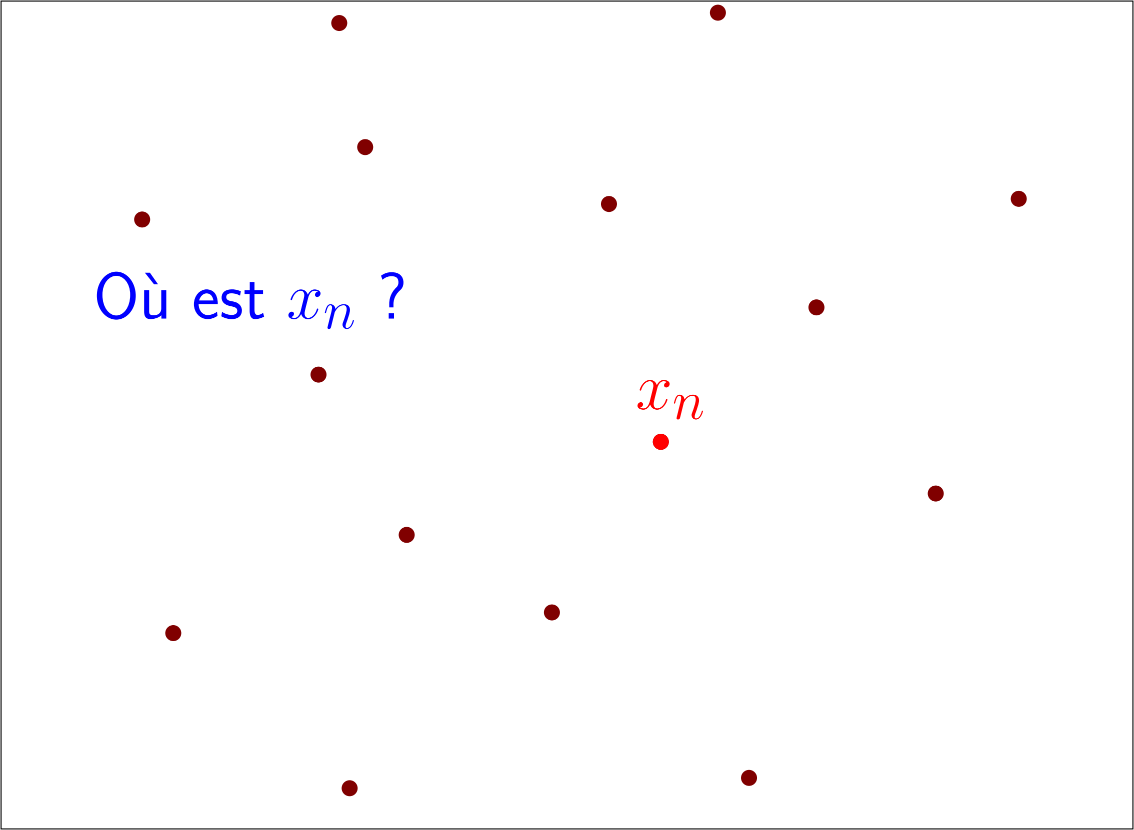
Idée : décomposer sur la

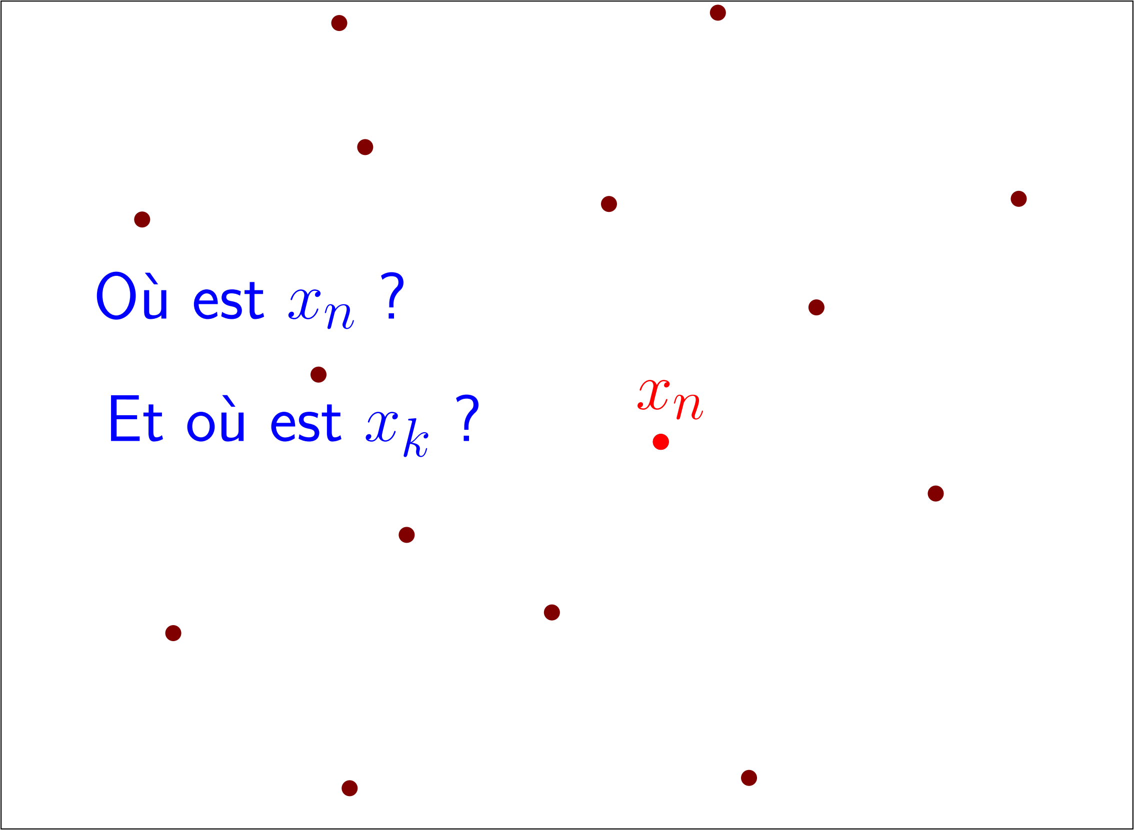
création de t

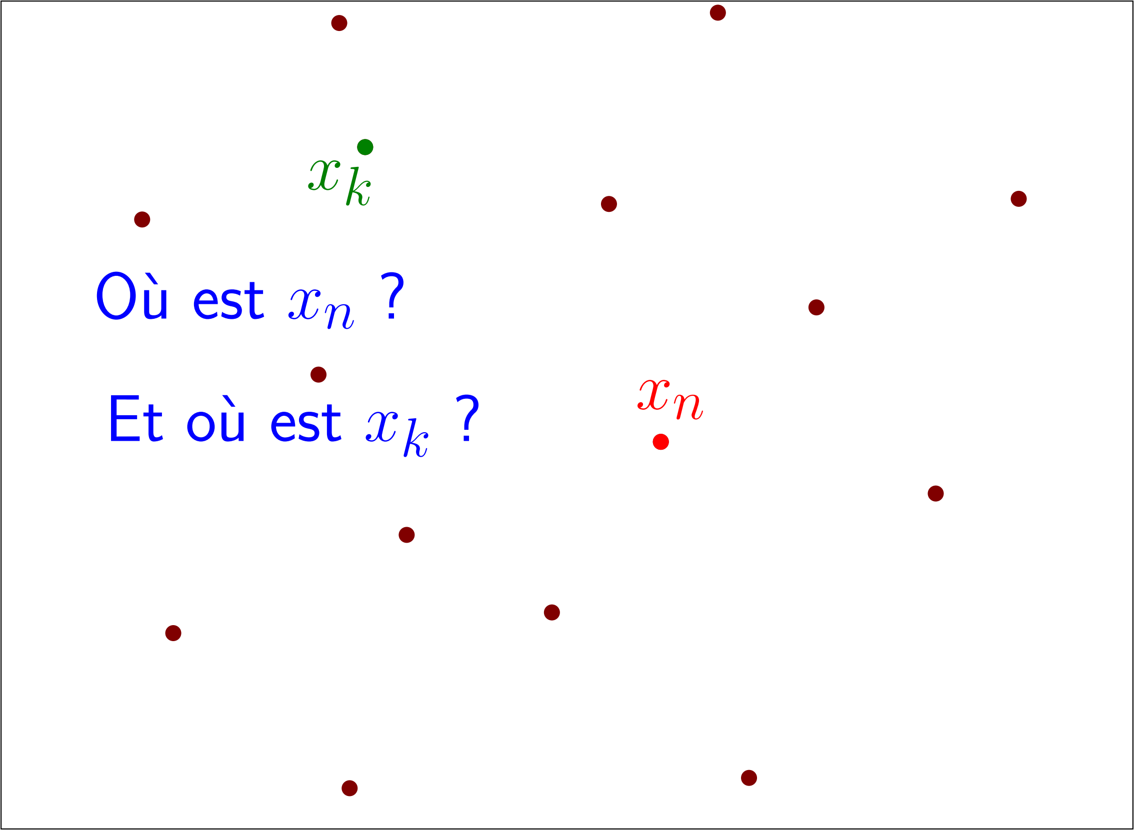

n points

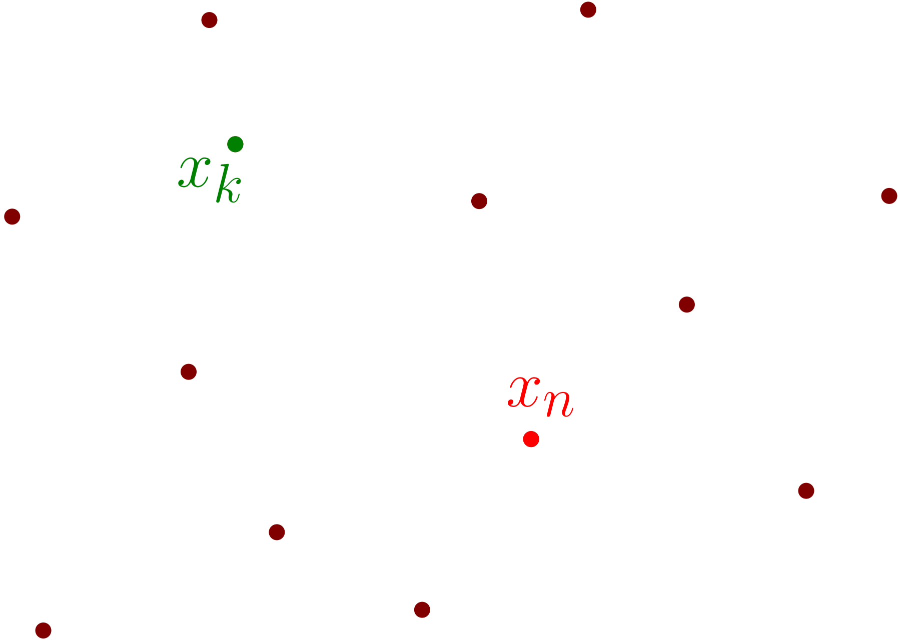

n points
échantillon $k+1$

Où est x_n ?

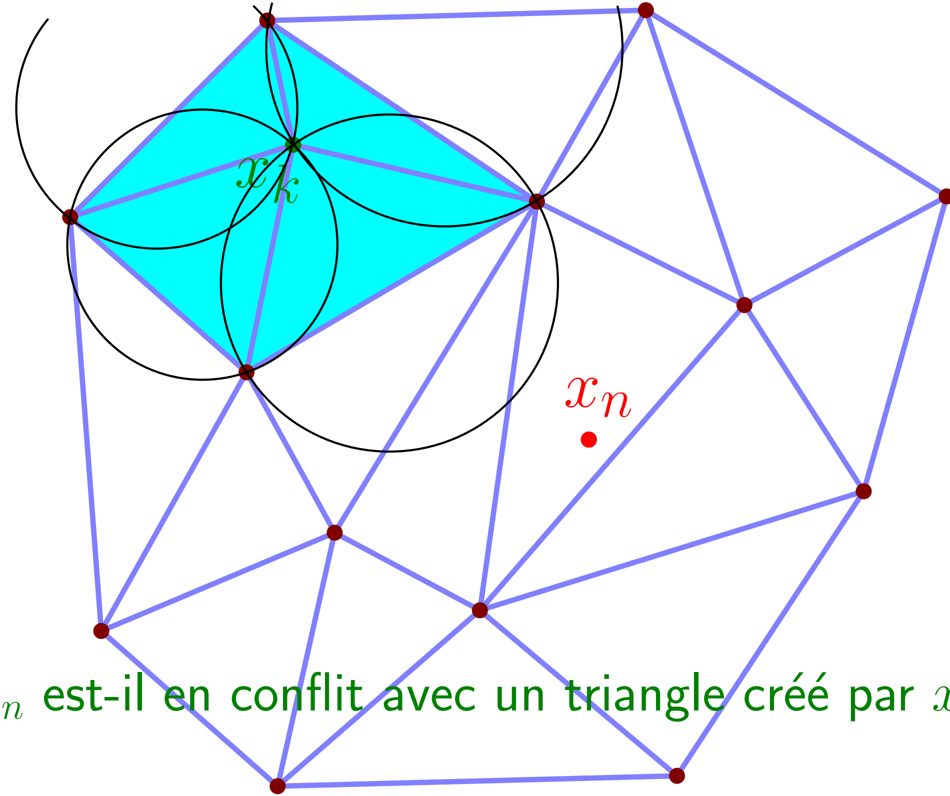

Où est x_n ?
x_n


Où est x_n ?
Et où est x_k ?
x_n

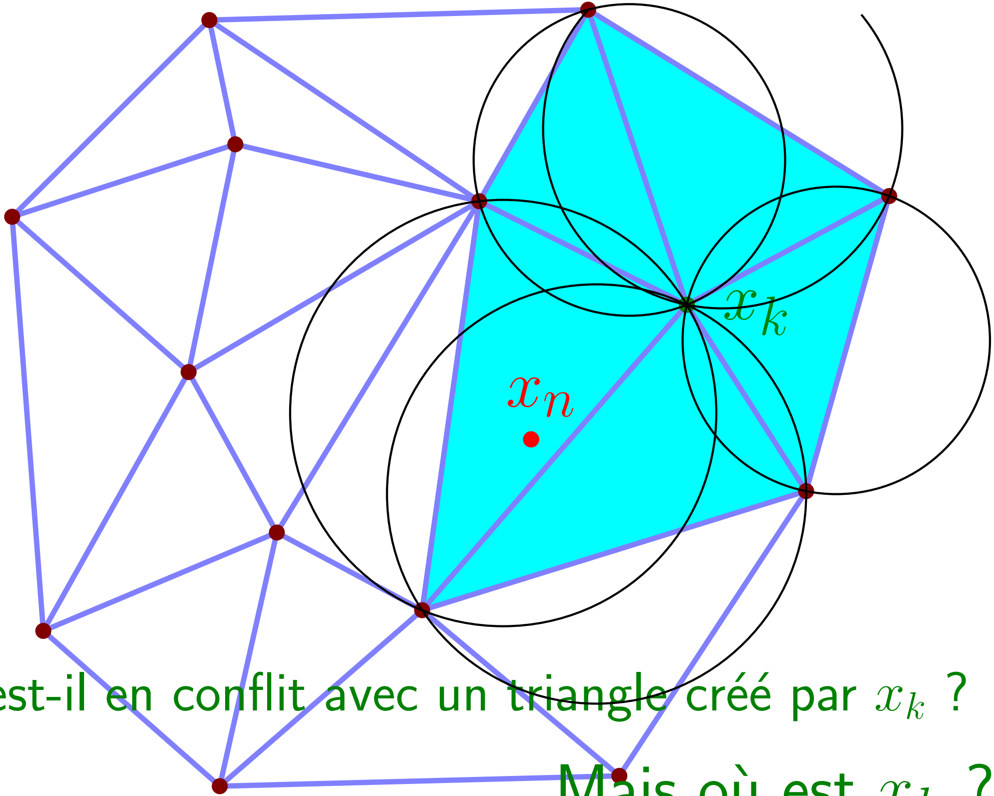

x_k
Où est x_n ?
Et où est x_k ?
x_n



x_n est-il en conflit avec un triangle créé par x_k ?

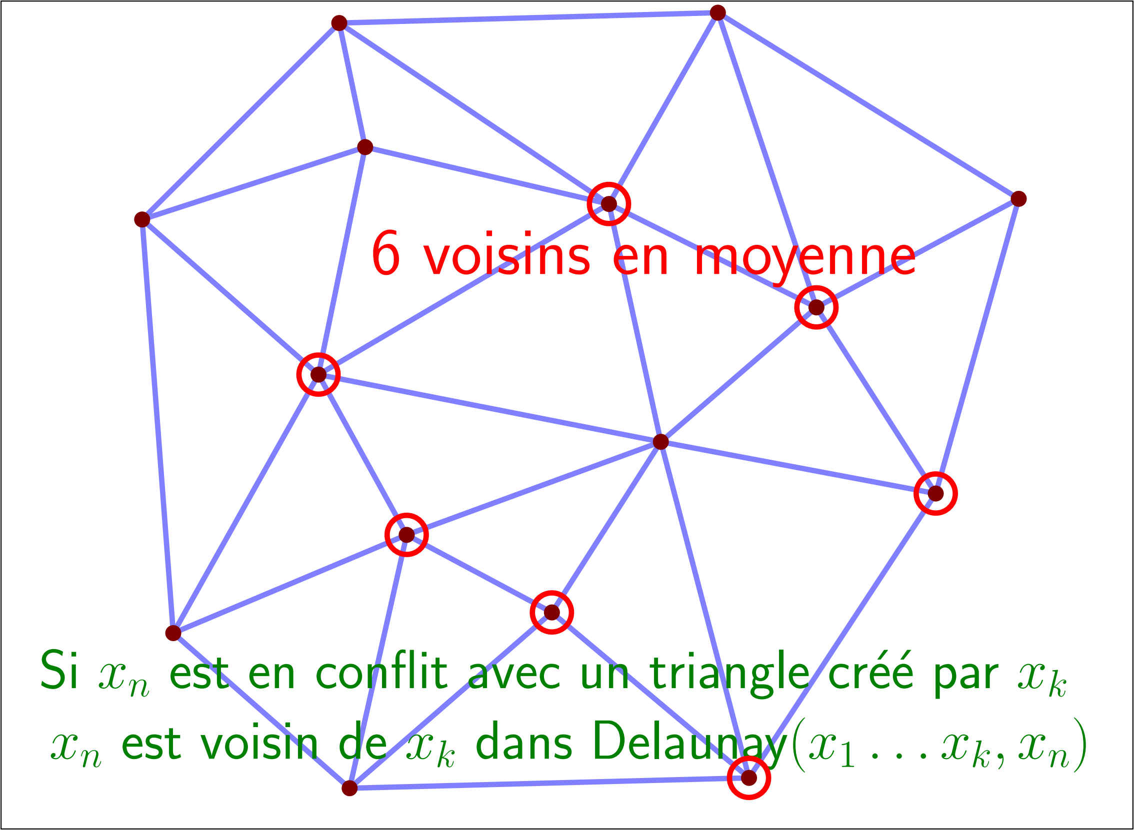


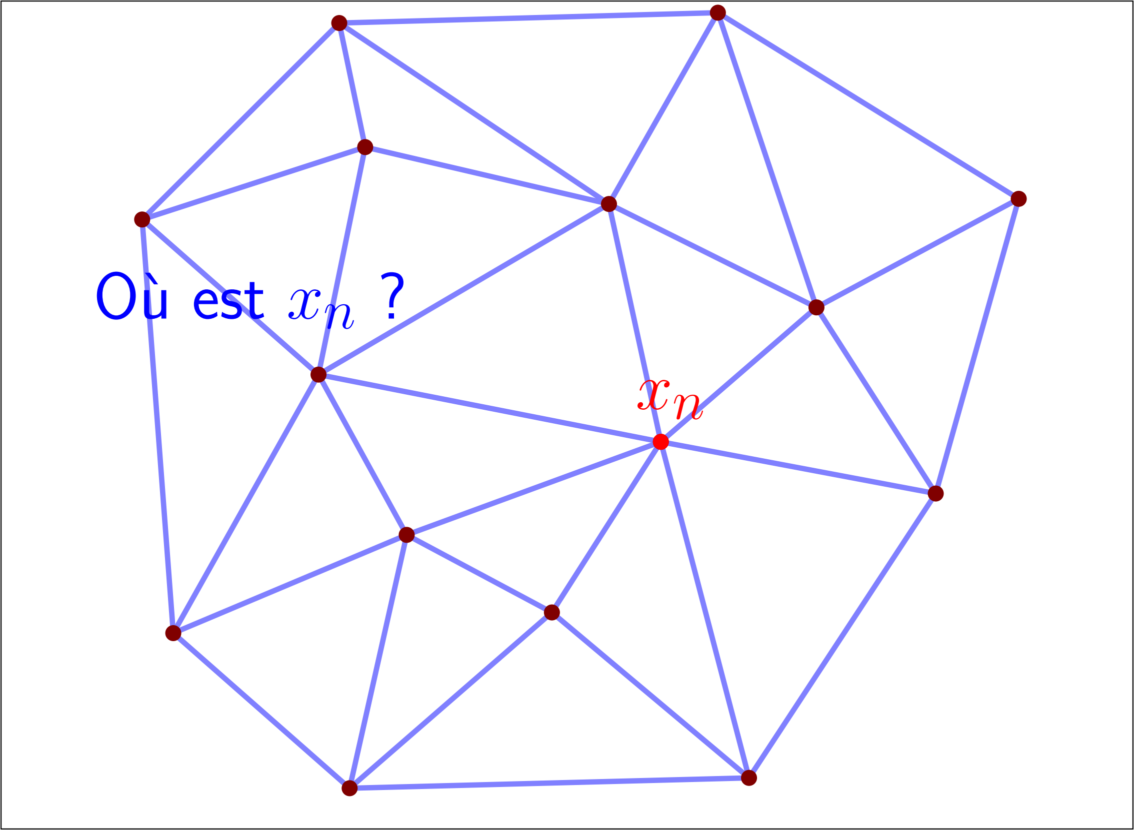
x_n est-il en conflit avec un triangle créé par x_k ?

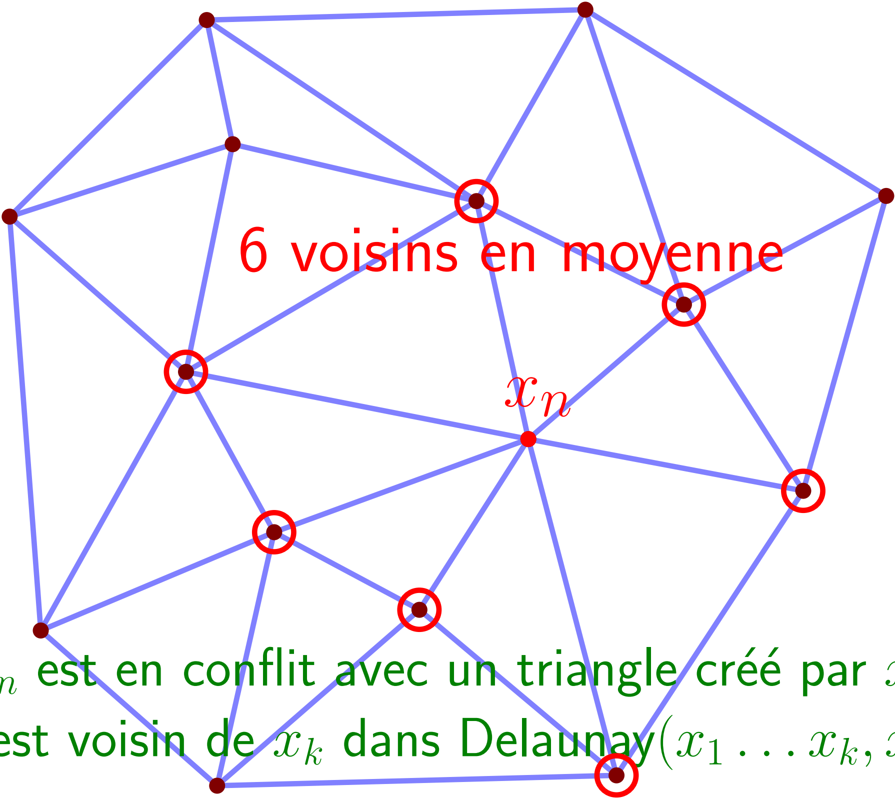


x_n est-il en conflit avec un triangle créé par x_k ?

Mais où est x_k ?

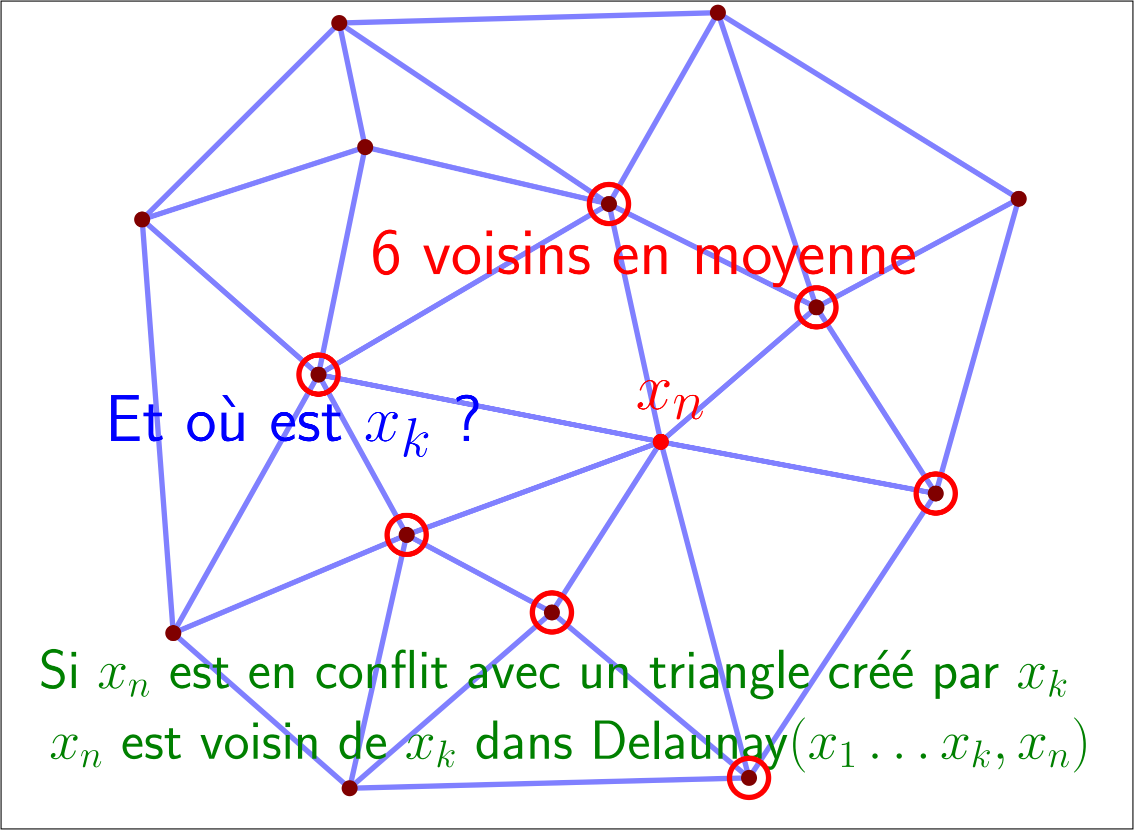

6 voisins en moyenne
Si x_n est en conflit avec un triangle créé par x_k
x_n est voisin de x_k dans Delaunay$(x_1 \ldots x_k, x_n)$


Où est x_n ?
x_n

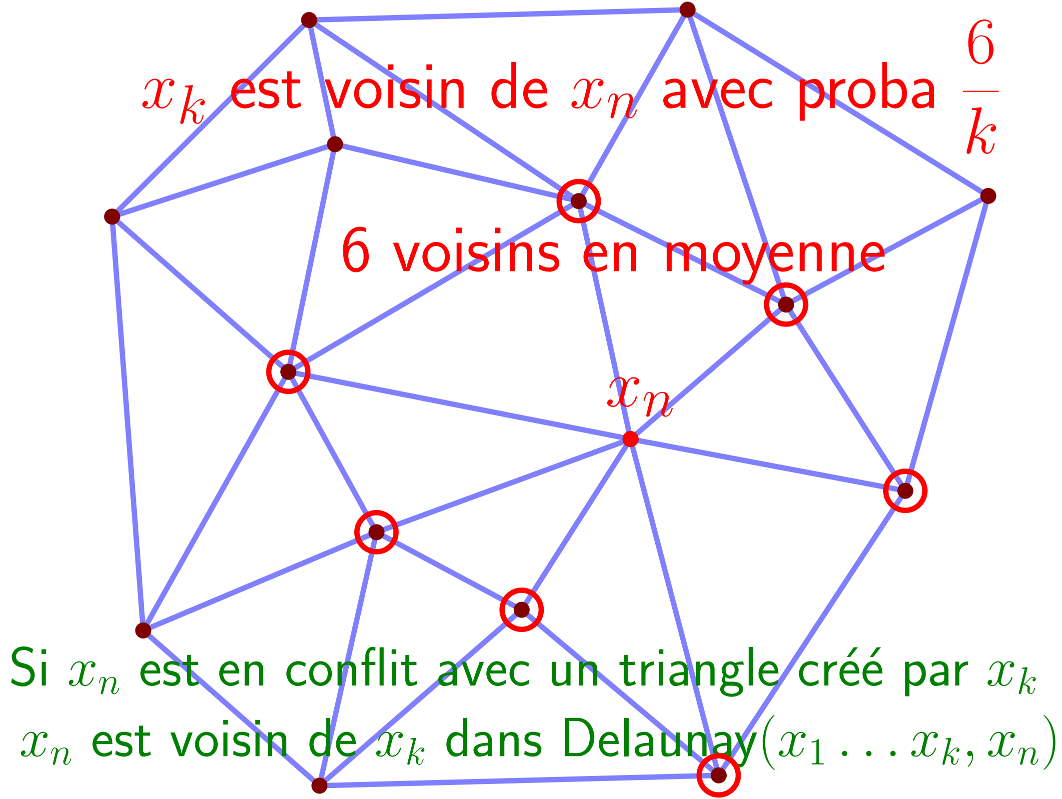


Si x_n est en conflit avec un triangle créé par x_k

x_n est voisin de x_k dans Delaunay$(x_1 \ldots x_k, x_n)$



6 voisins en moyenne
Et où est x_k ?
x_n
Si x_n est en conflit avec un triangle créé par x_k
x_n est voisin de x_k dans Delaunay$(x_1 \ldots x_k, x_n)$



x_k est voisin de x_n avec proba $\frac{6}{k}$
6 voisins en moyenne
x_n
Si x_n est en conflit avec un triangle créé par x_k
x_n est voisin de x_k dans Delaunay$(x_1 \ldots x_k, x_n)$

On teste les conflits entre x_n

et un triangle créé par x_k

avec proba $\dfrac{6}{k}$

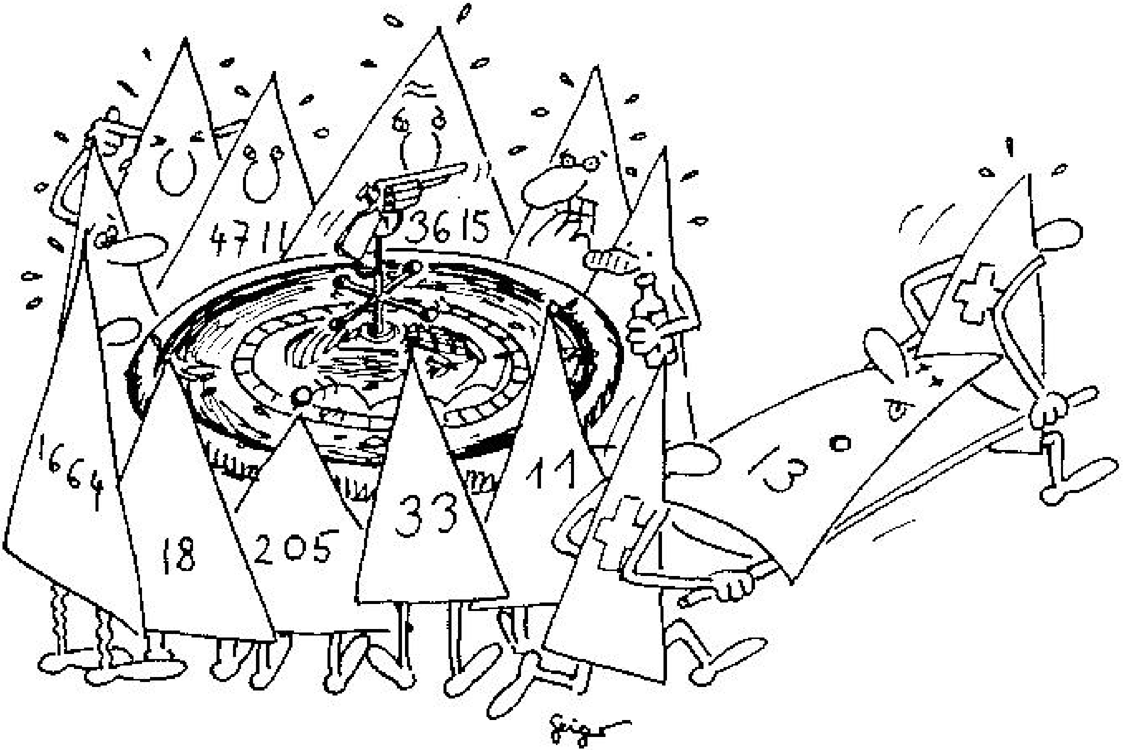
Algorithme incrémental

Coût dernier point ?

localisation

$$\sum_k \frac{6}{k} = \log n$$

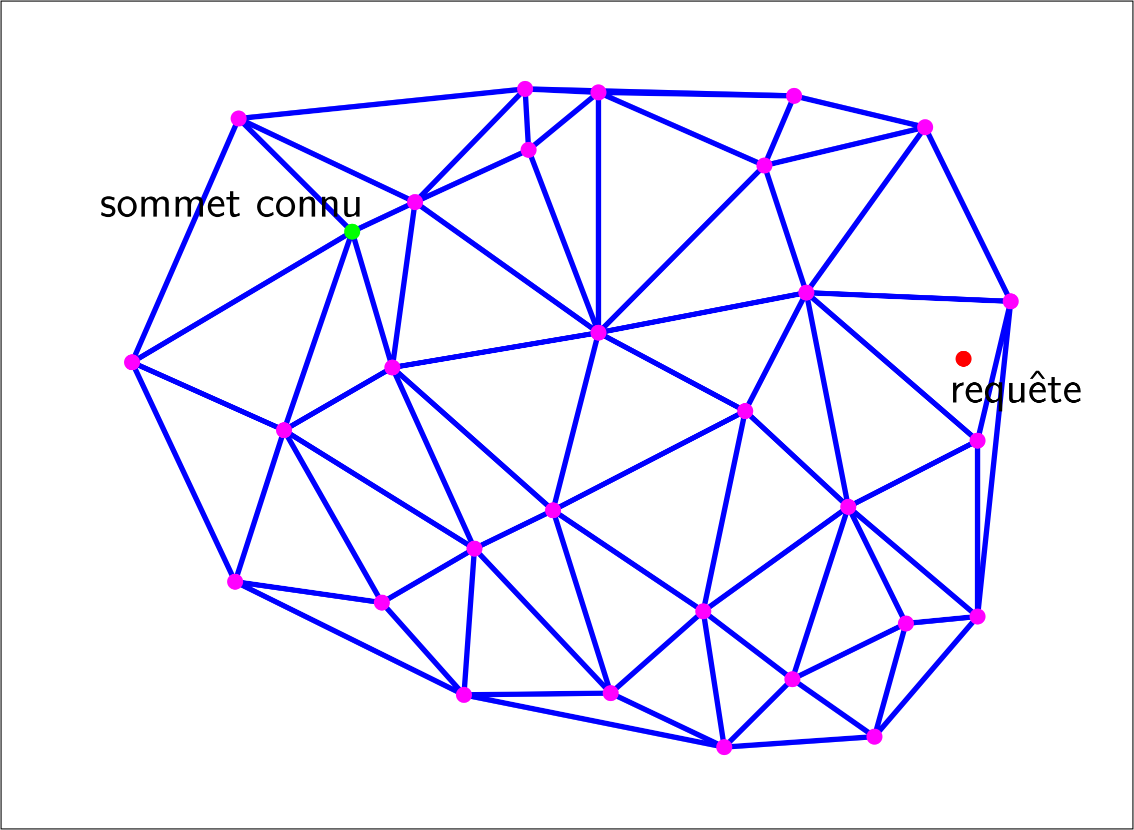
$$O(n \log n)$$

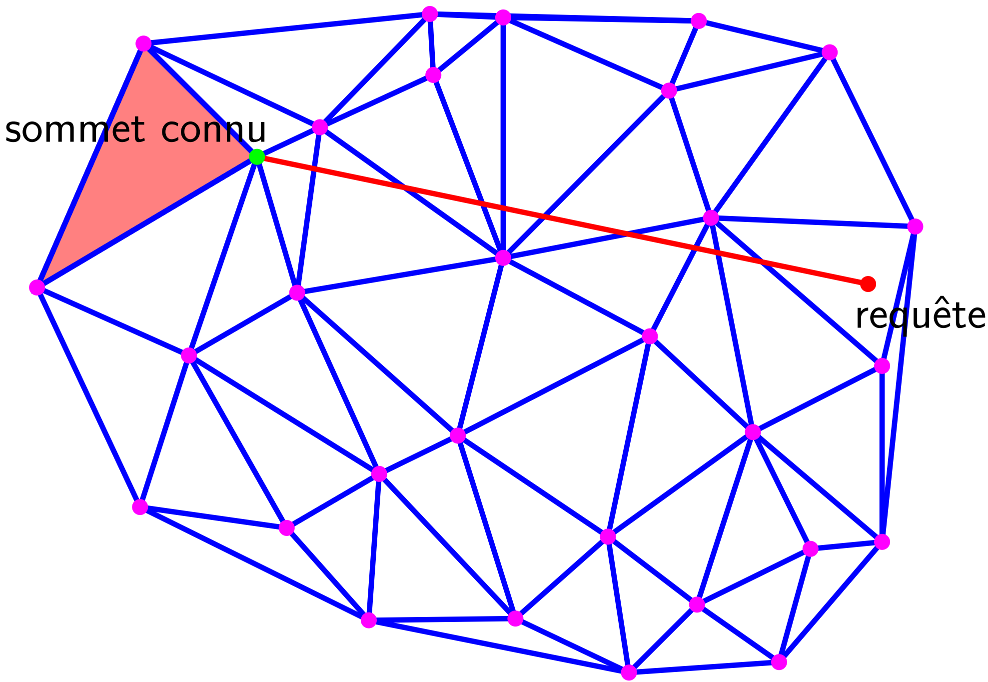

4711
3615
1664
18
205
33
11
13

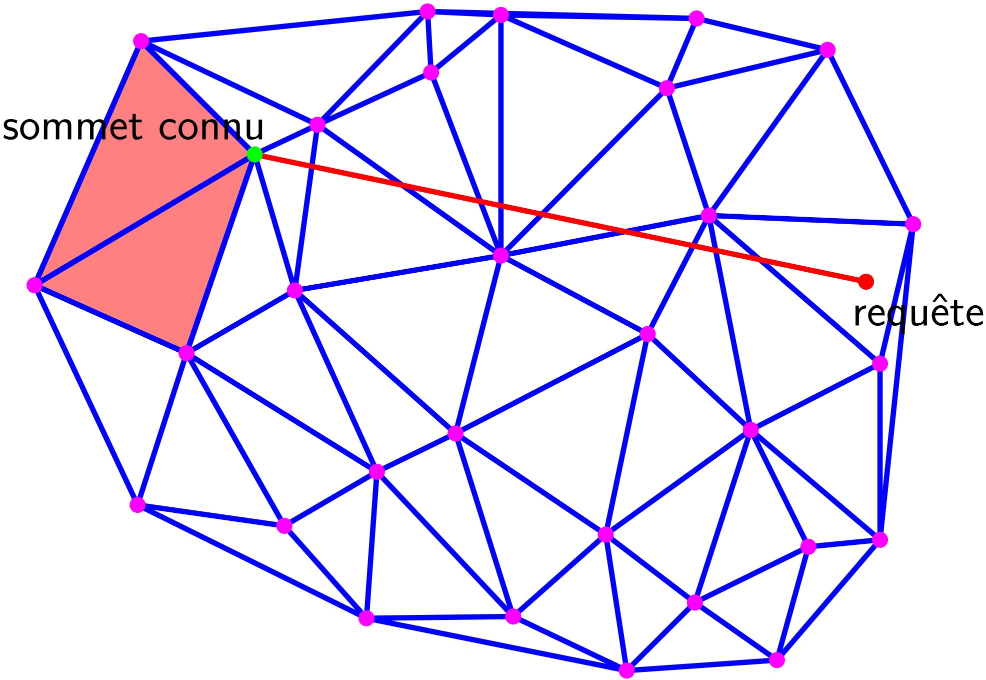
Marcher dans

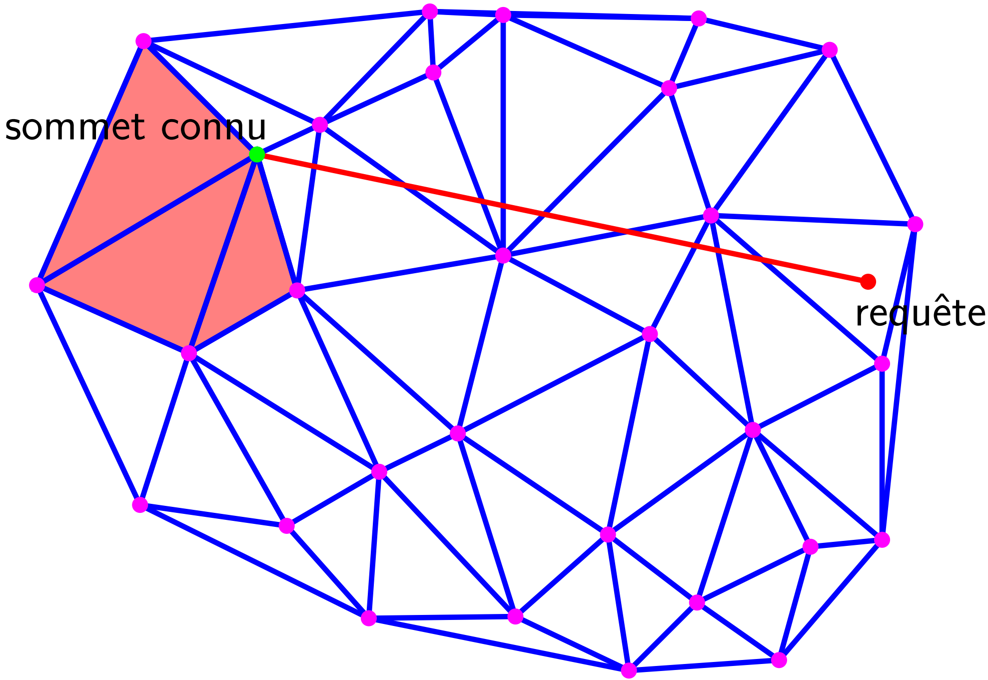
la triangulation de Delaunay

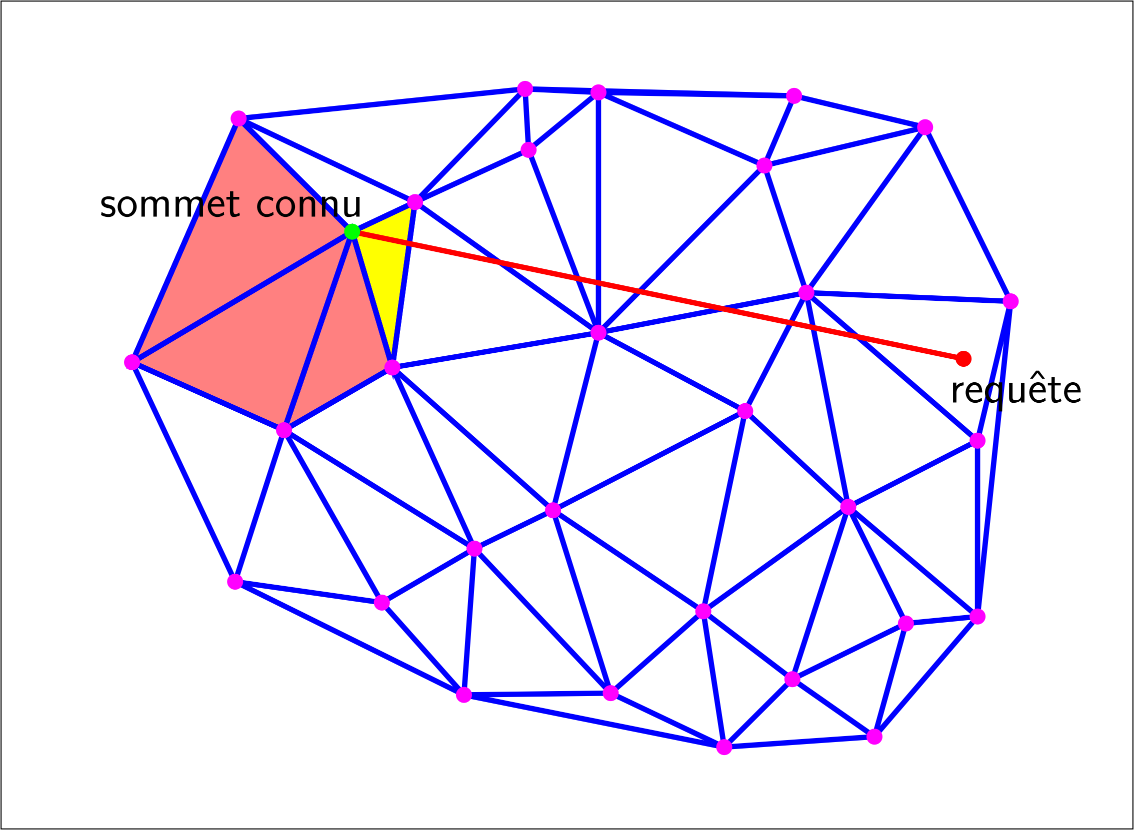
Marche rectiligne

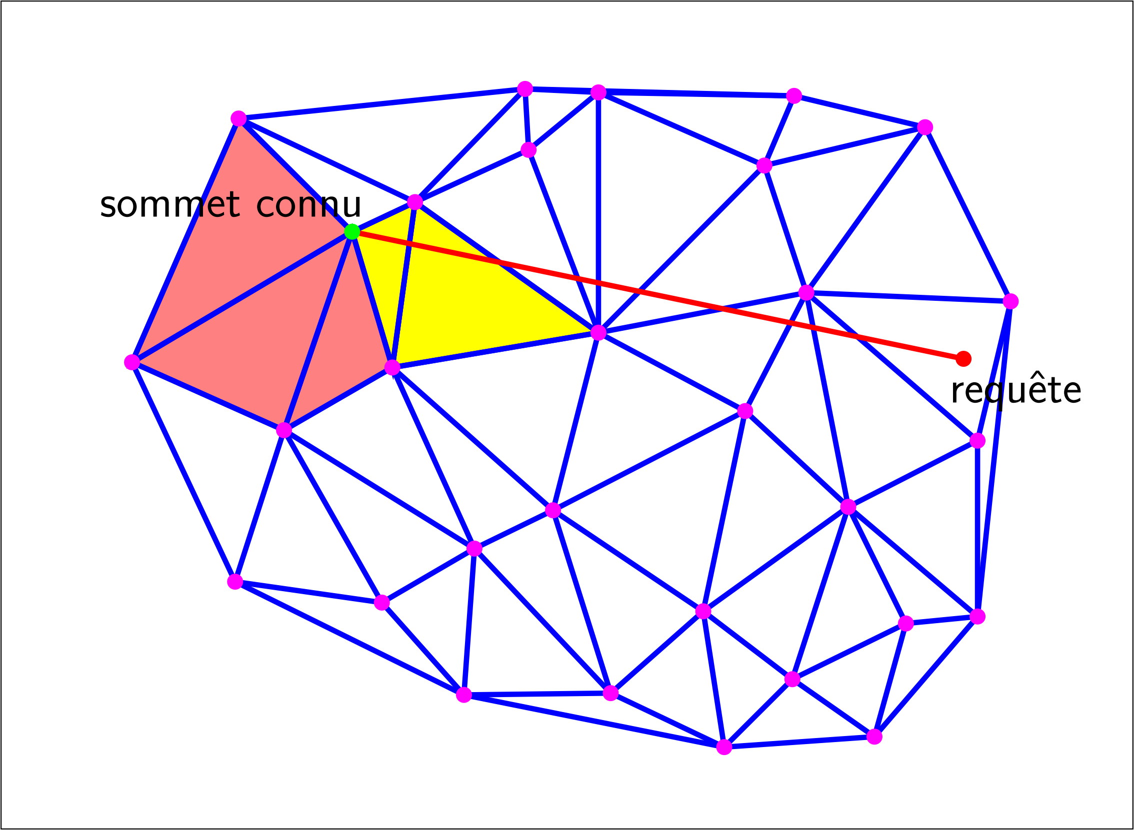

sommet connu
requête

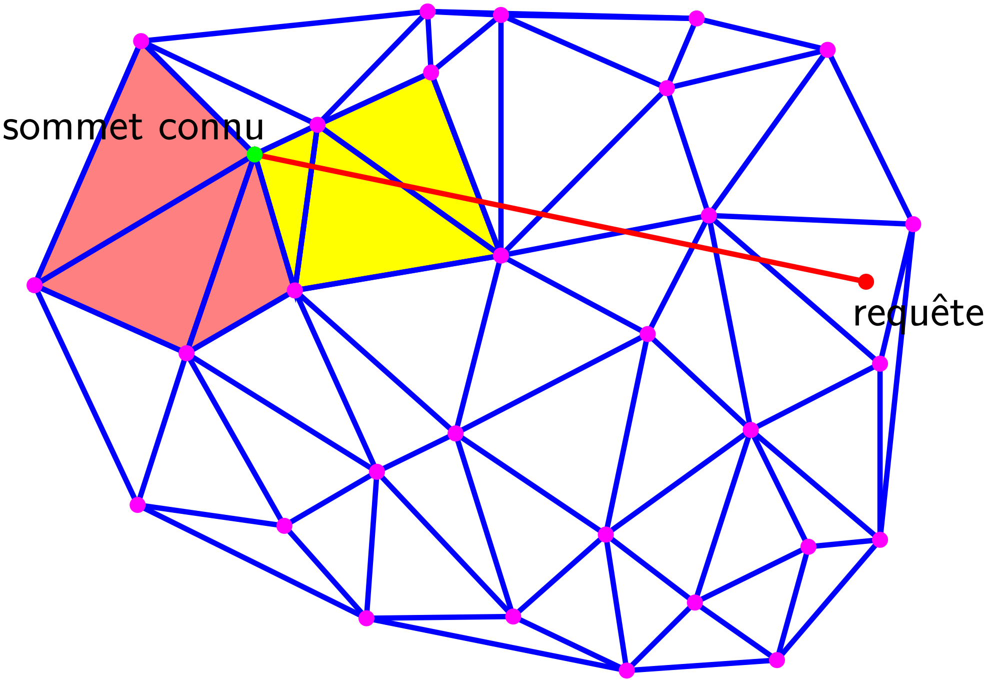

sommet connu
requête

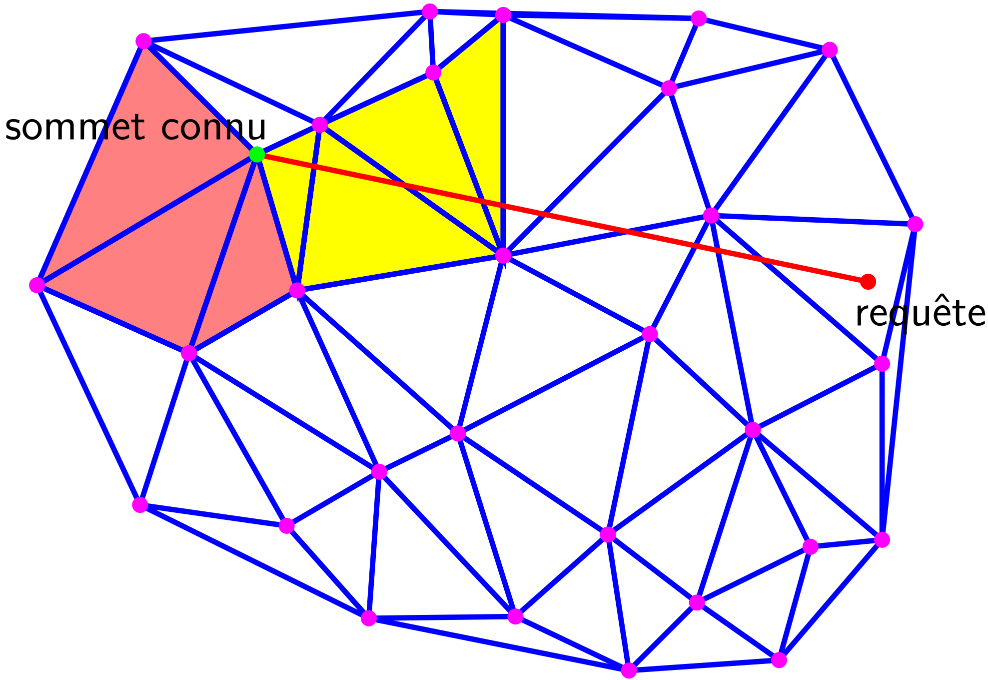

sommet connu
requête

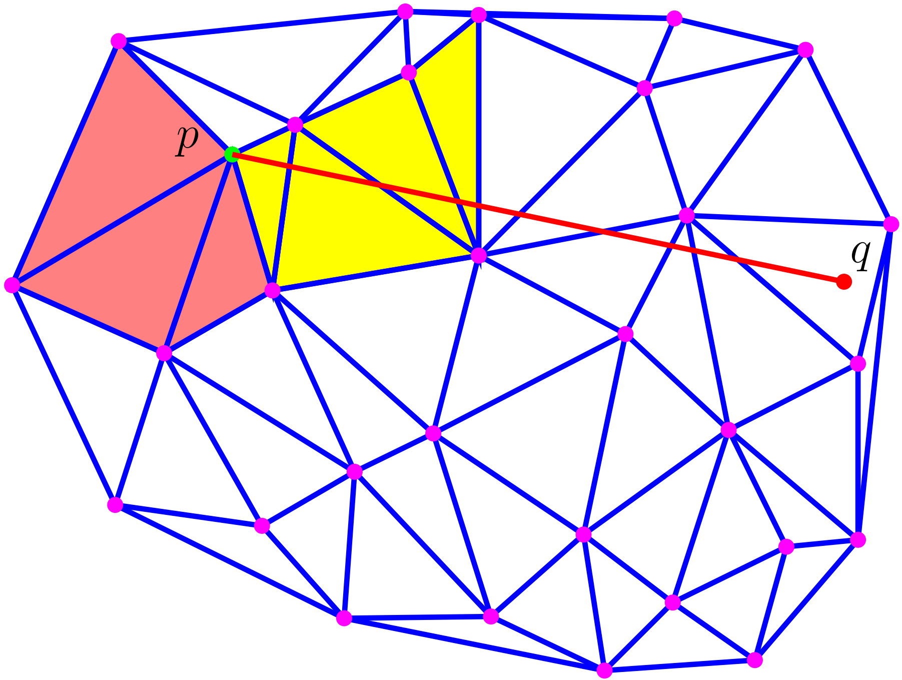

sommet connu
requête


sommet connu
requête

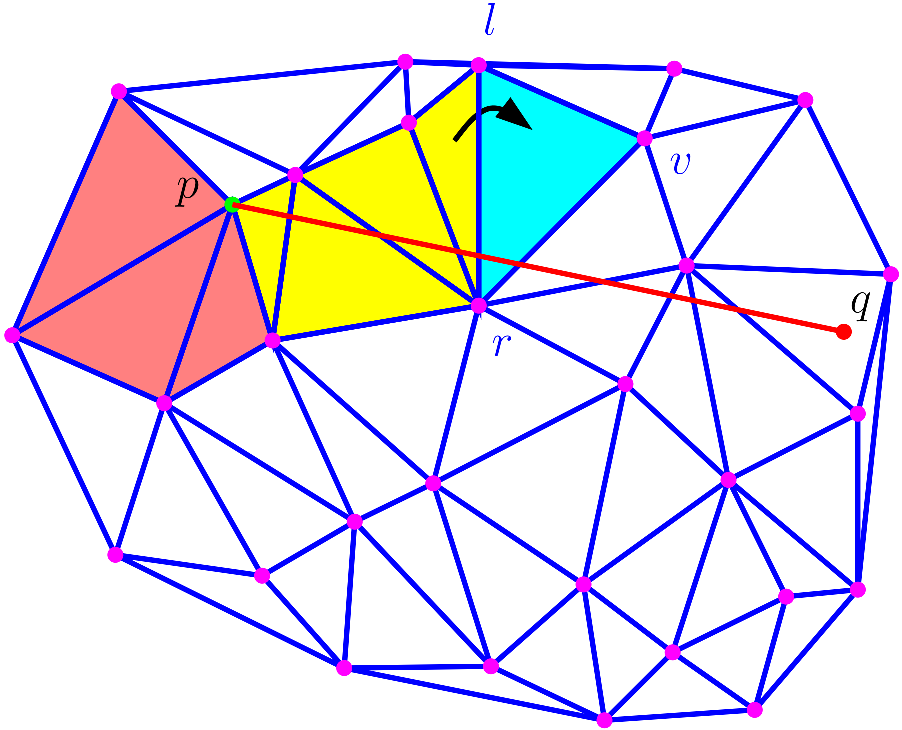

sommet connu
requête

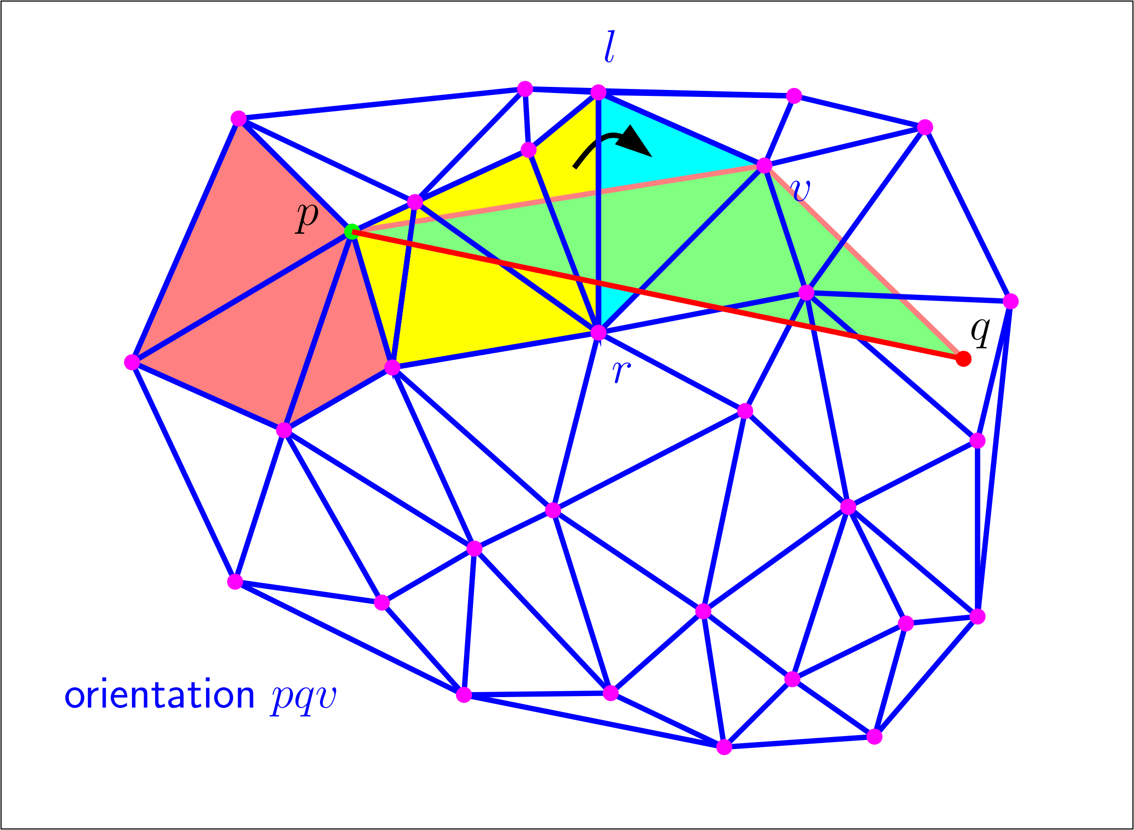

sommet connu
requête

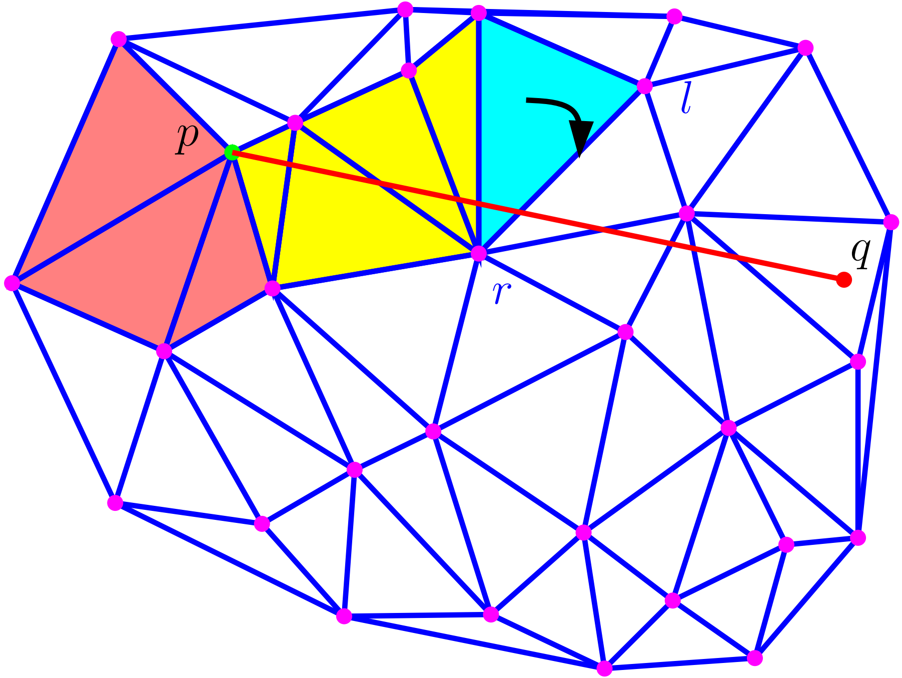

sommet connu
requête



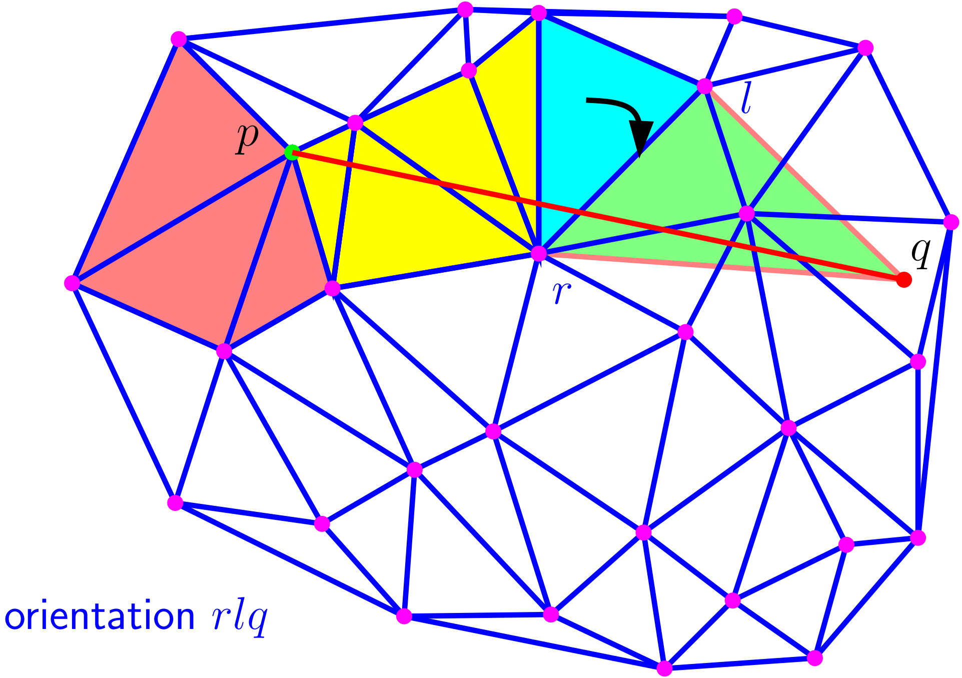
p
q

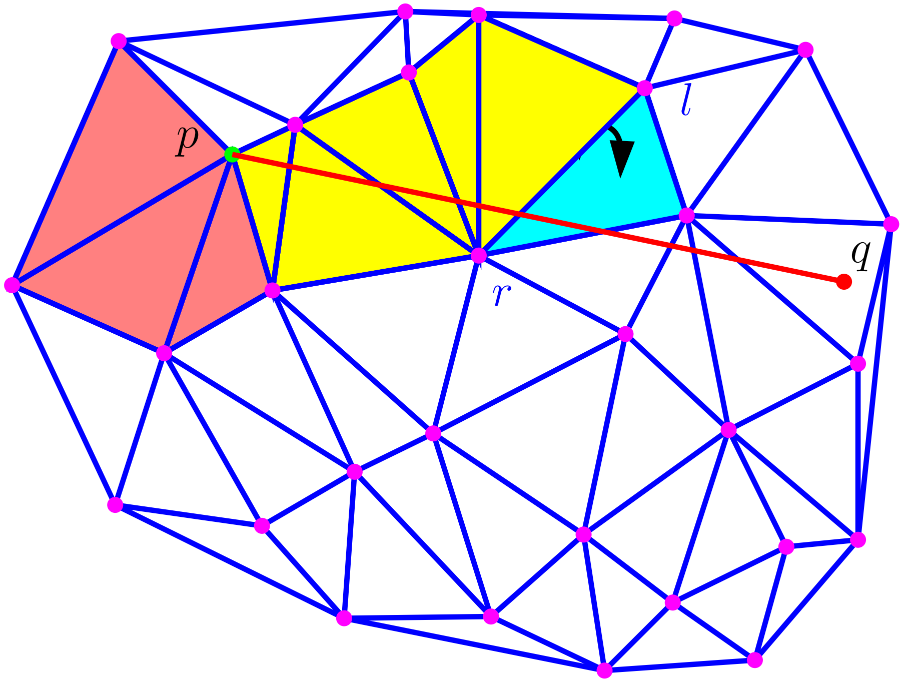

l
v
p
q
r

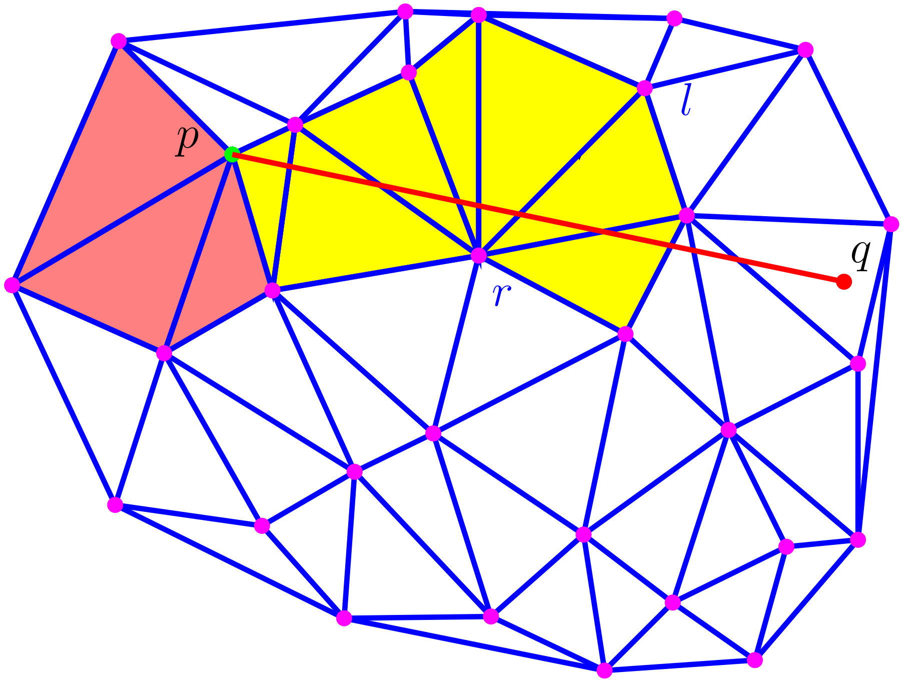

l
p
v
q
r
orientation pqv

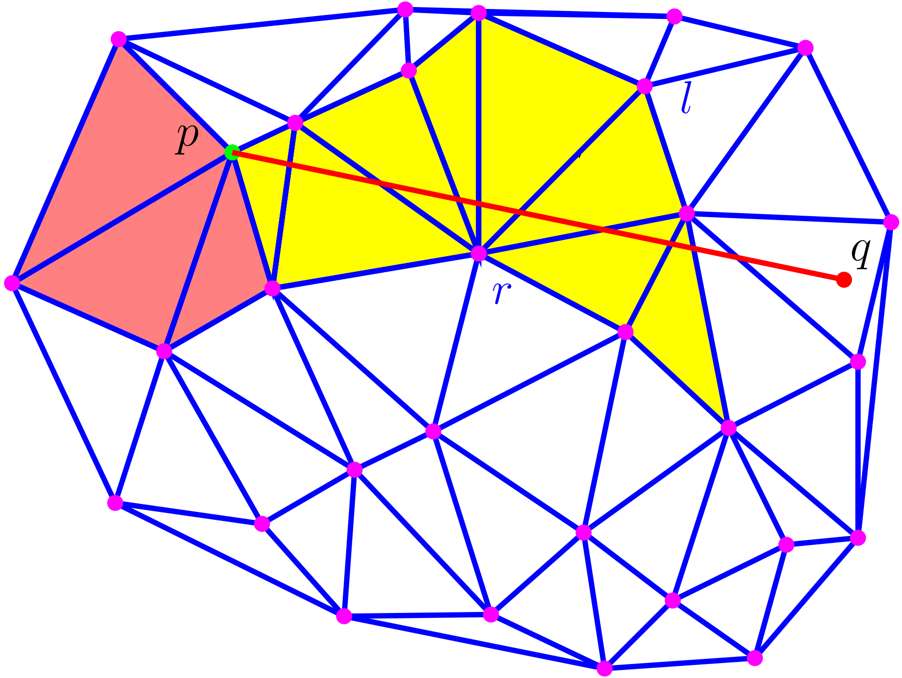


p
l
q
r

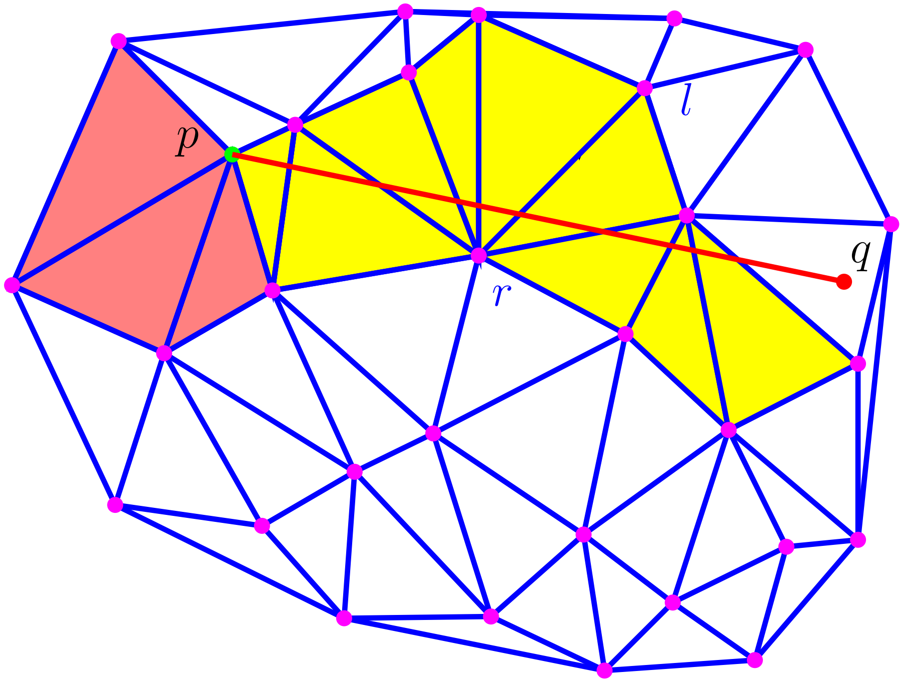

p
l
q
r
orientation rlq


p
l
q
r

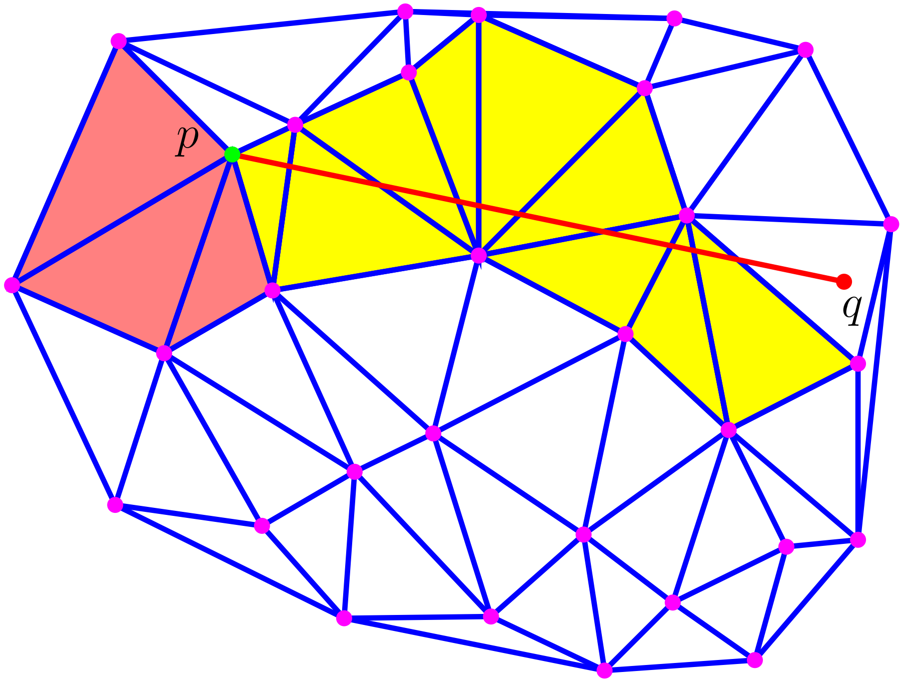

p
l
q
r



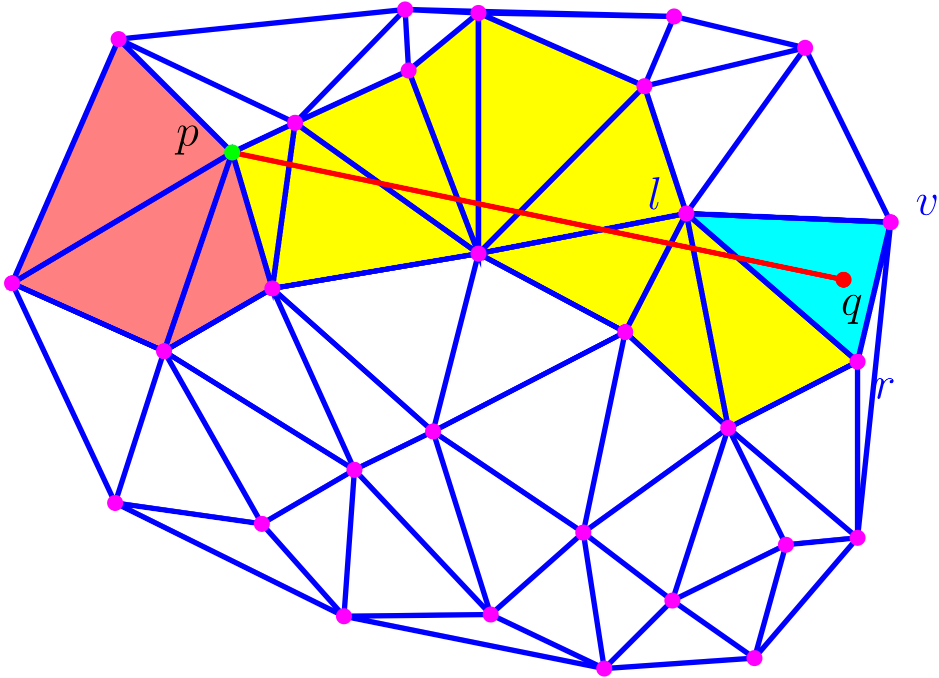
p
l
q
r

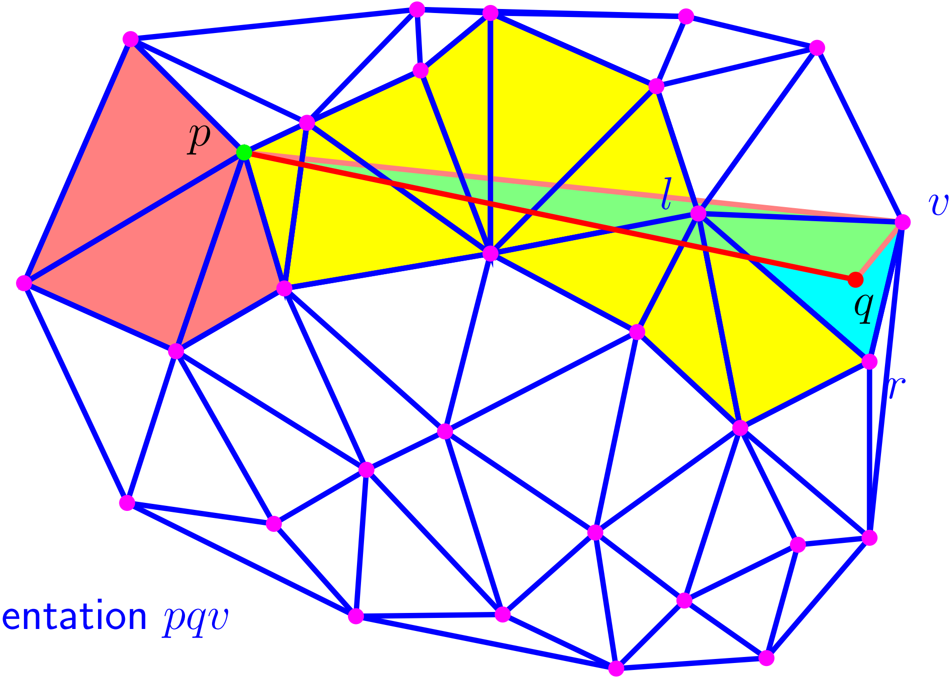


p
l
q
r

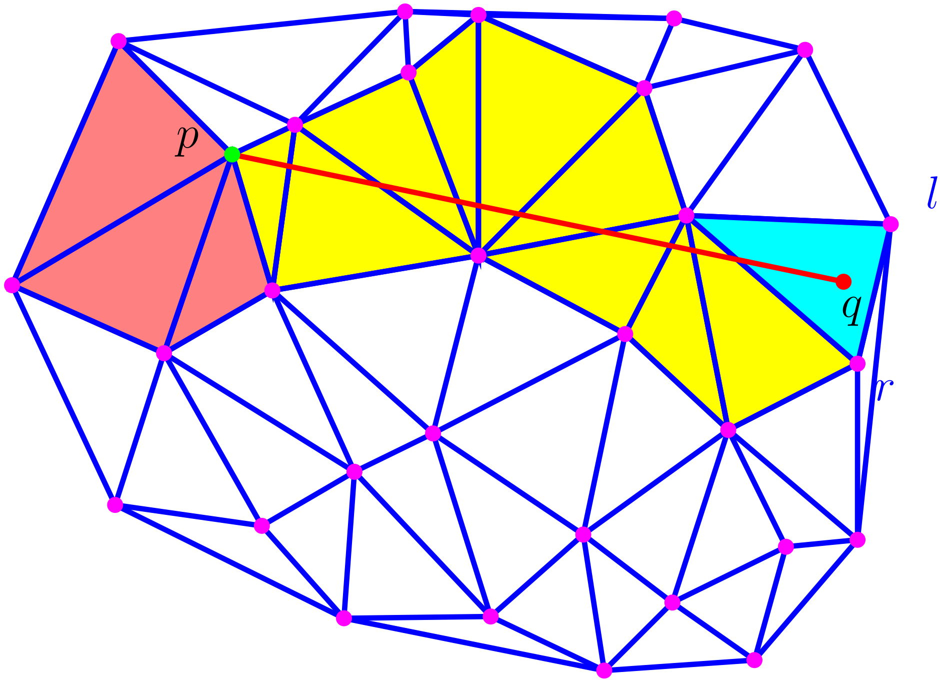


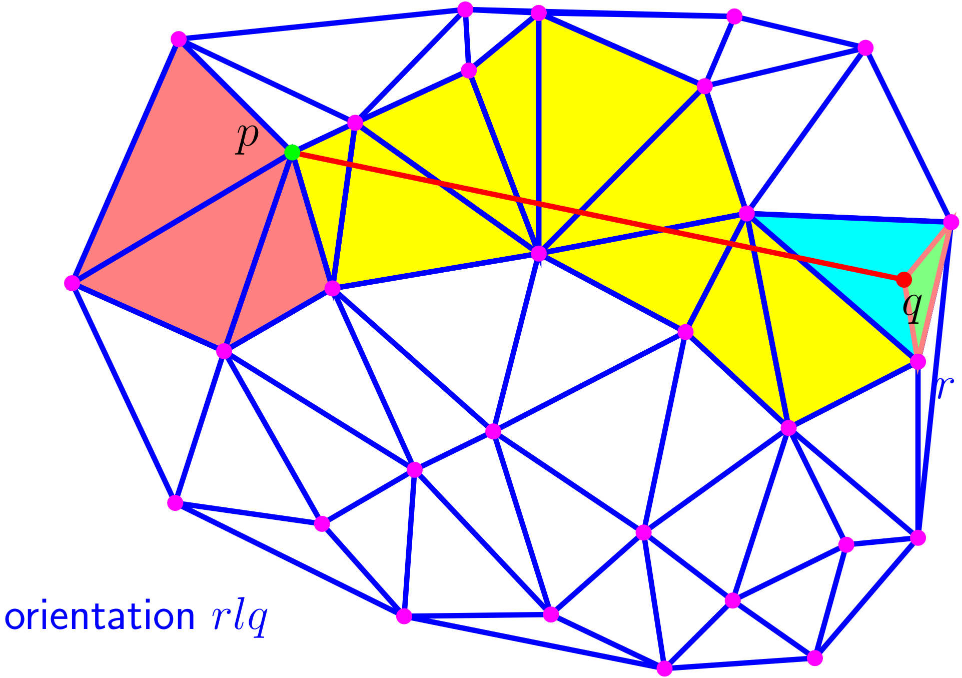
p
q


p
l
v
q
r



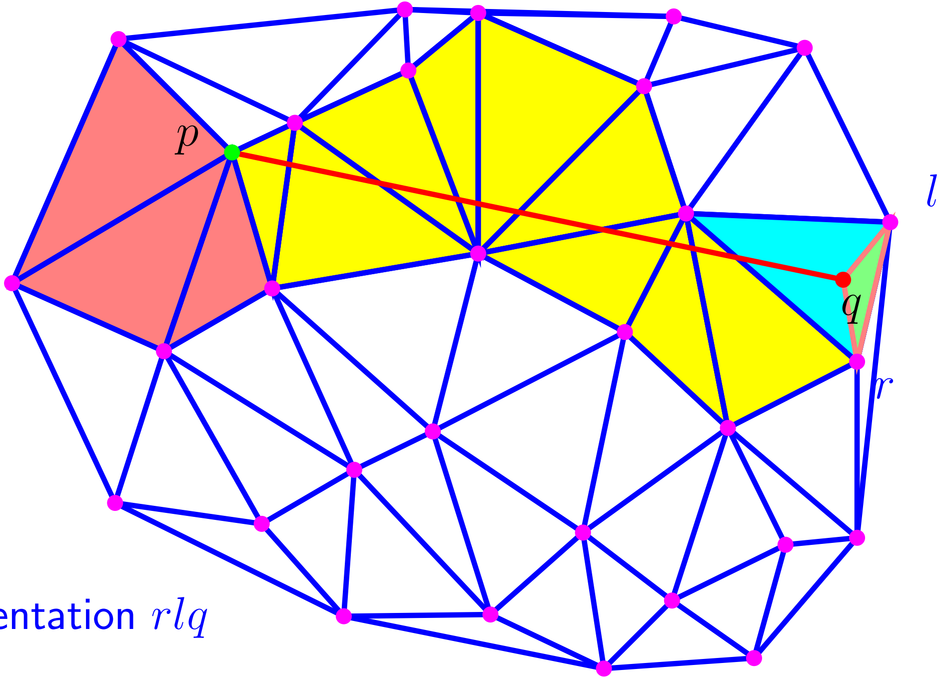
orientation pqv


p
q
l
r



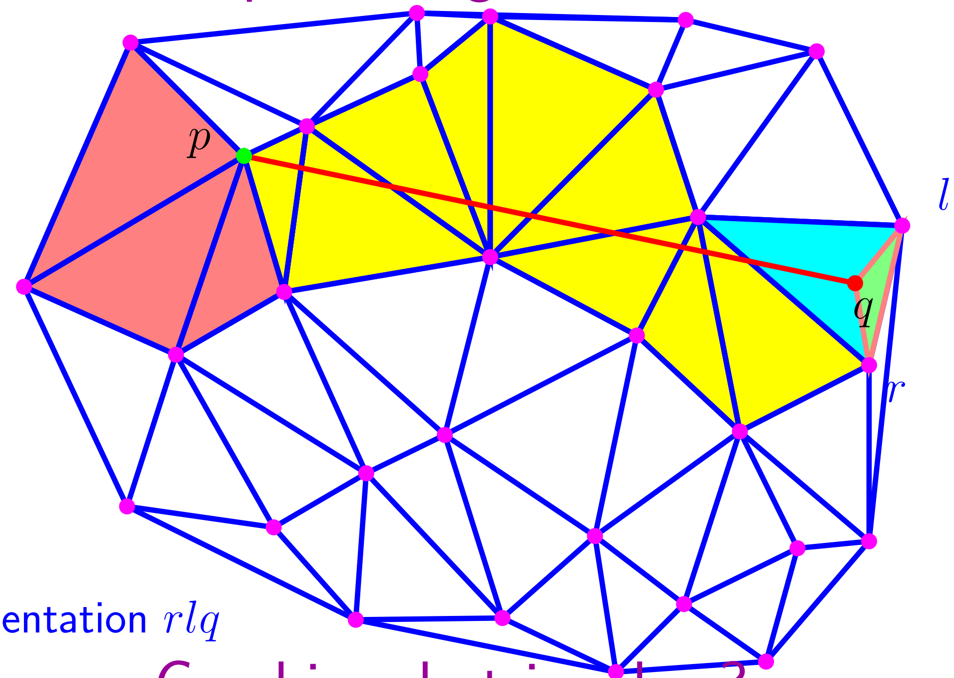
p
q
l
r
orientation rlq

Deux tests par triangles



orientation rlq

Deux tests par triangles



orientation rlq

Combien de triangles ?

Combien de triangles ?

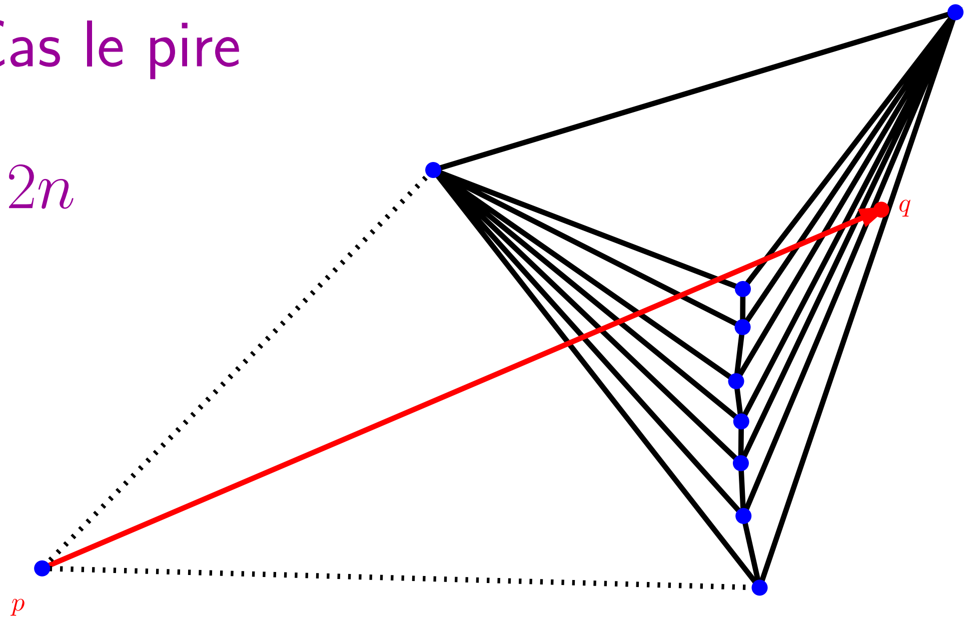
Combien de triangles ?

Cas le pire

Combien de triangles ?

Cas le pire

$\simeq 2n$



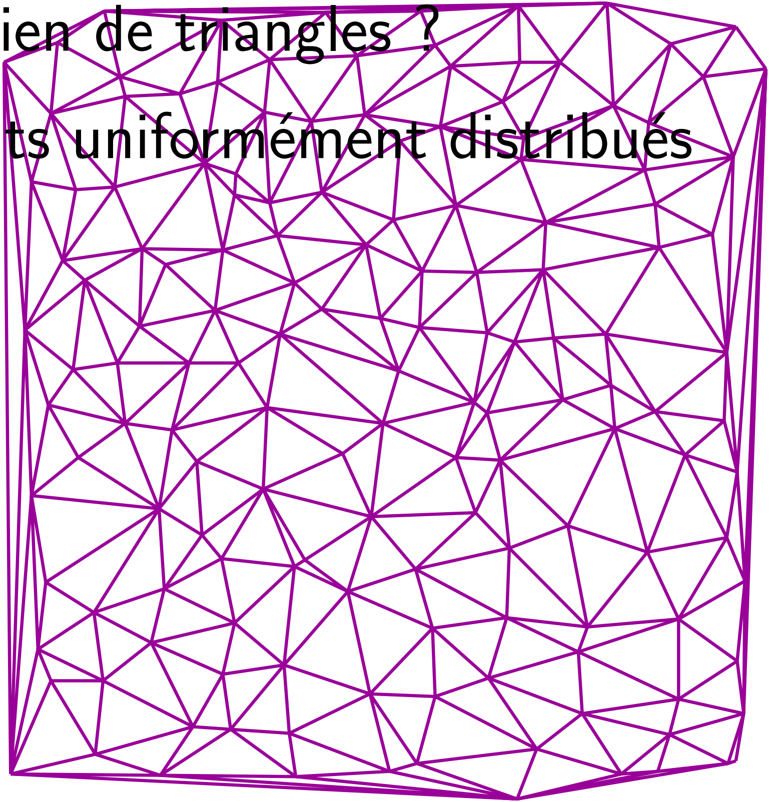
Combien de triangles ?

Combien de triangles ?

Points uniformément distribués

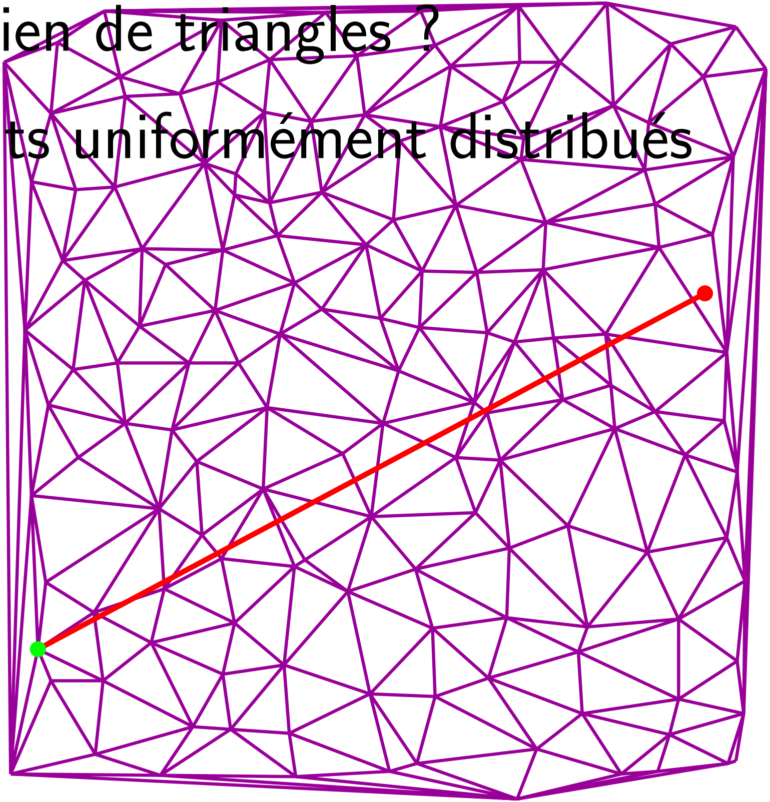
Combien de triangles ?

Points uniformément distribués



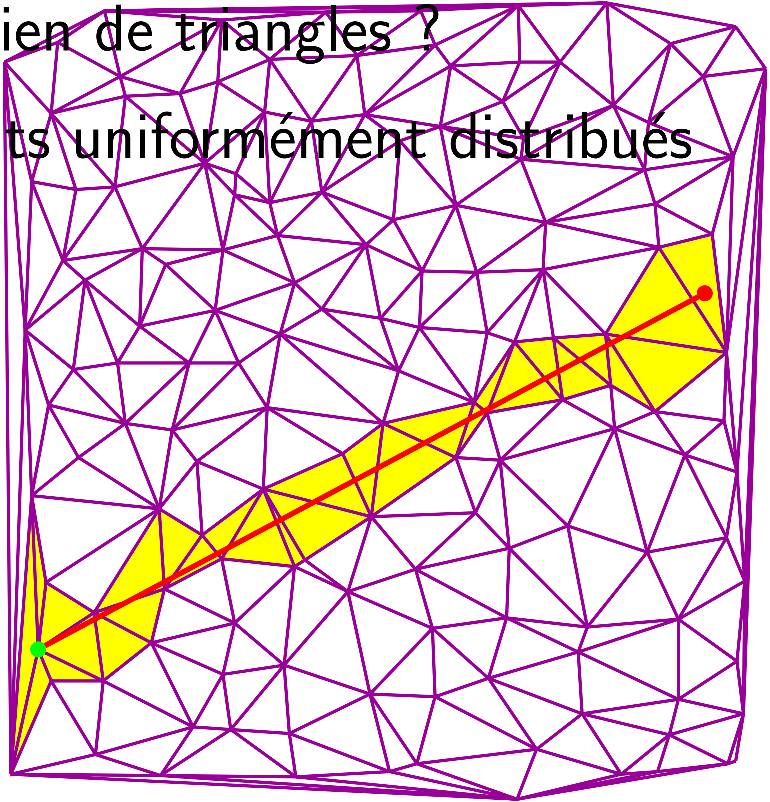
Combien de triangles ?

Points uniformément distribués



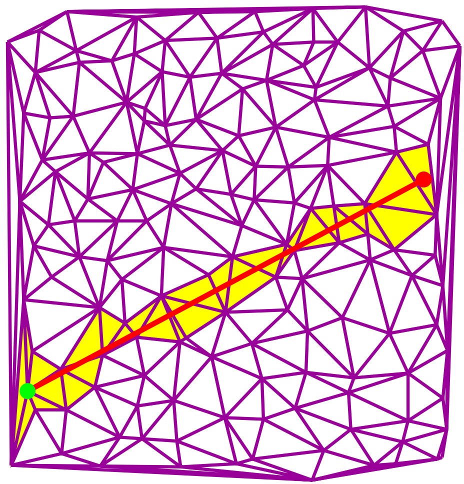
Combien de triangles ?

Points uniformément distribués



Combien de triangles ?

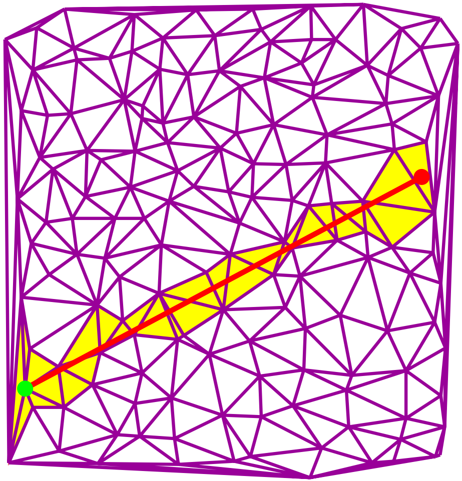
Points uniformément distribués



Combien de triangles ?

Points uniformément distribués

Intuition



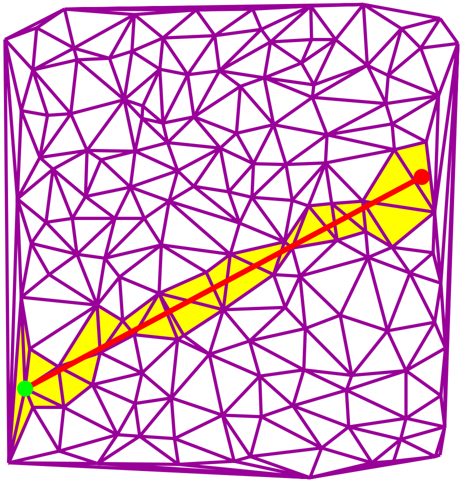
Combien de triangles ?

Points uniformément distribués

Intuition

Aire d'un triangle :

Largeur d'un triangle :



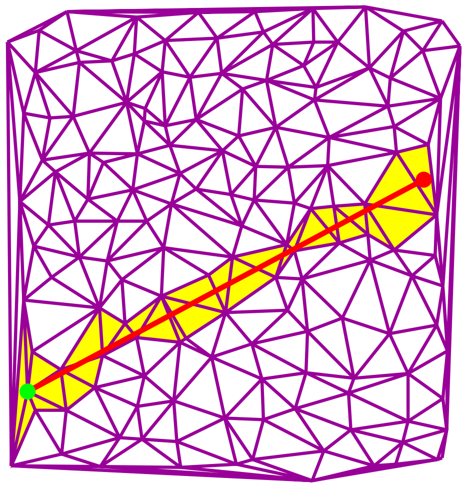
Combien de triangles ?

Points uniformément distribués

Intuition

Aire d'un triangle : $\simeq \frac{1}{2n}$

Largeur d'un triangle : $\simeq \frac{1}{\sqrt{n}}$



Combien de triangles ?

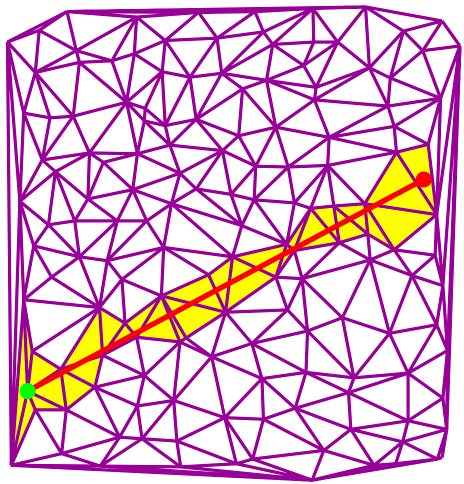
Points uniformément distribués

Intuition

Aire d'un triangle : $\simeq \frac{1}{2n}$

Largeur d'un triangle : $\simeq \frac{1}{\sqrt{n}}$

Aire visitée



Combien de triangles ?

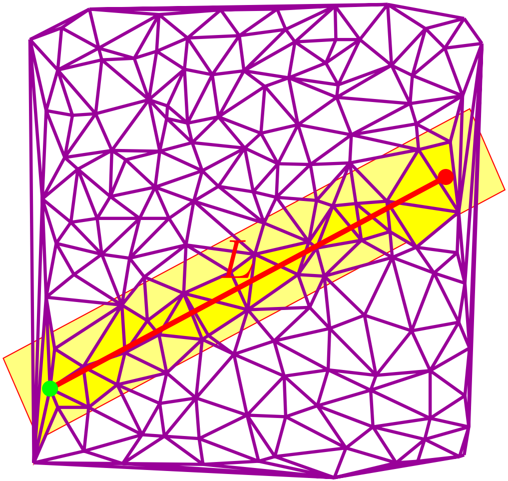
Points uniformément distribués

Intuition

Aire d'un triangle : $\simeq \frac{1}{2n}$

Largeur d'un triangle : $\simeq \frac{1}{\sqrt{n}}$

Aire visitée $\simeq L \times \frac{1}{\sqrt{n}}$



Combien de triangles ?

Points uniformément distribués

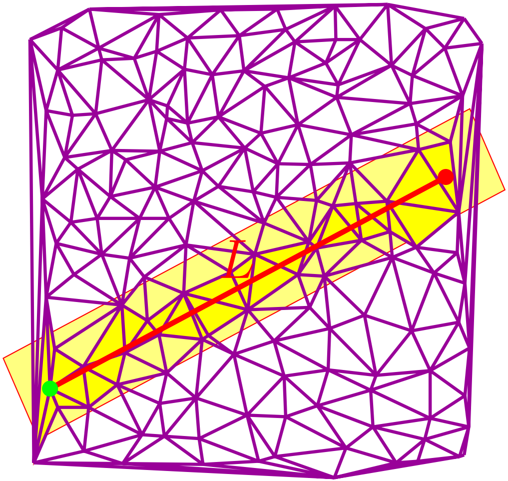
Intuition

Aire d'un triangle : $\simeq \frac{1}{2n}$

Largeur d'un triangle : $\simeq \frac{1}{\sqrt{n}}$

Aire visitée $\simeq L \times \frac{1}{\sqrt{n}}$

Nombre de triangles visités



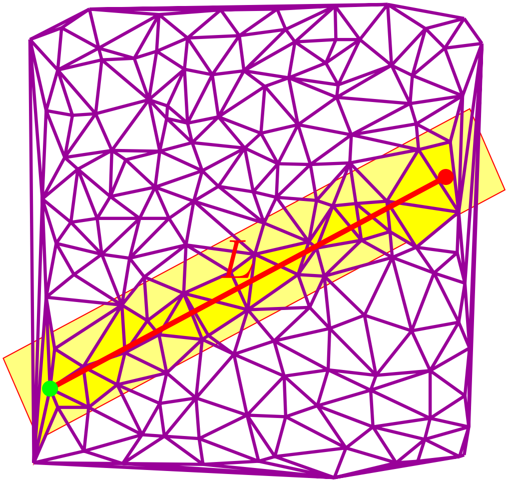
Combien de triangles ?

Points uniformément distribués

Intuition

Aire d'un triangle : $\simeq \frac{1}{2n}$

Largeur d'un triangle : $\simeq \frac{1}{\sqrt{n}}$



Aire visitée $\simeq L \times \frac{1}{\sqrt{n}}$

Nombre de triangles visités $\simeq \frac{L}{\sqrt{n}} \Big/ \frac{1}{2n} \simeq 2L\sqrt{n}$

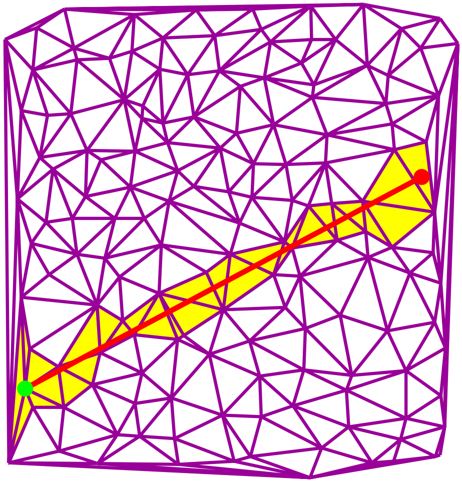
Combien de triangles ?

Points uniformément distribués

Intuition

Preuve théorique

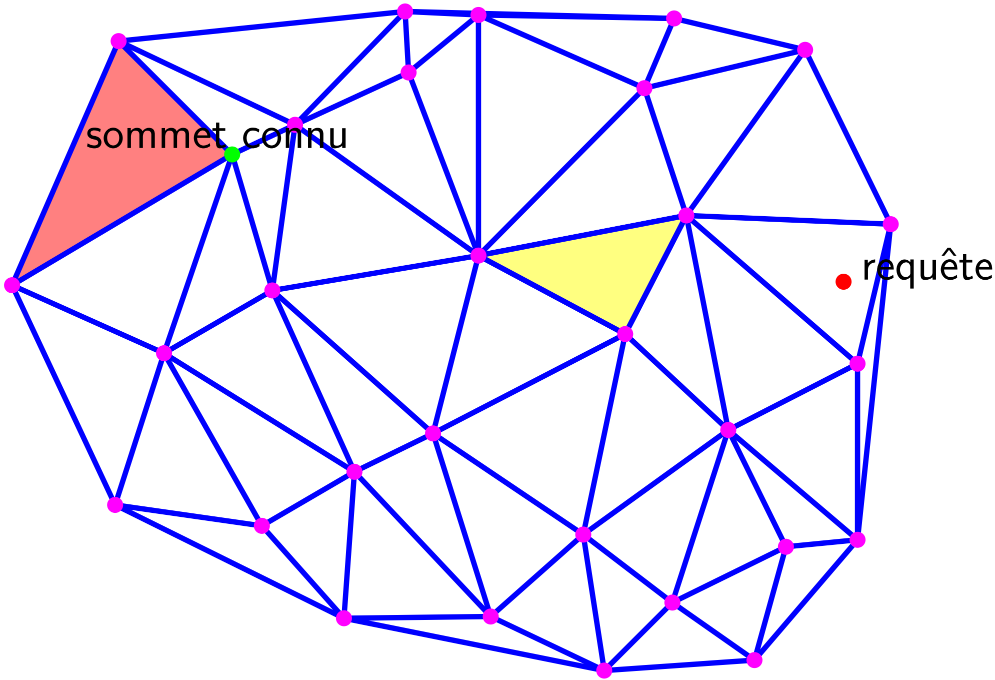
Plus difficile !

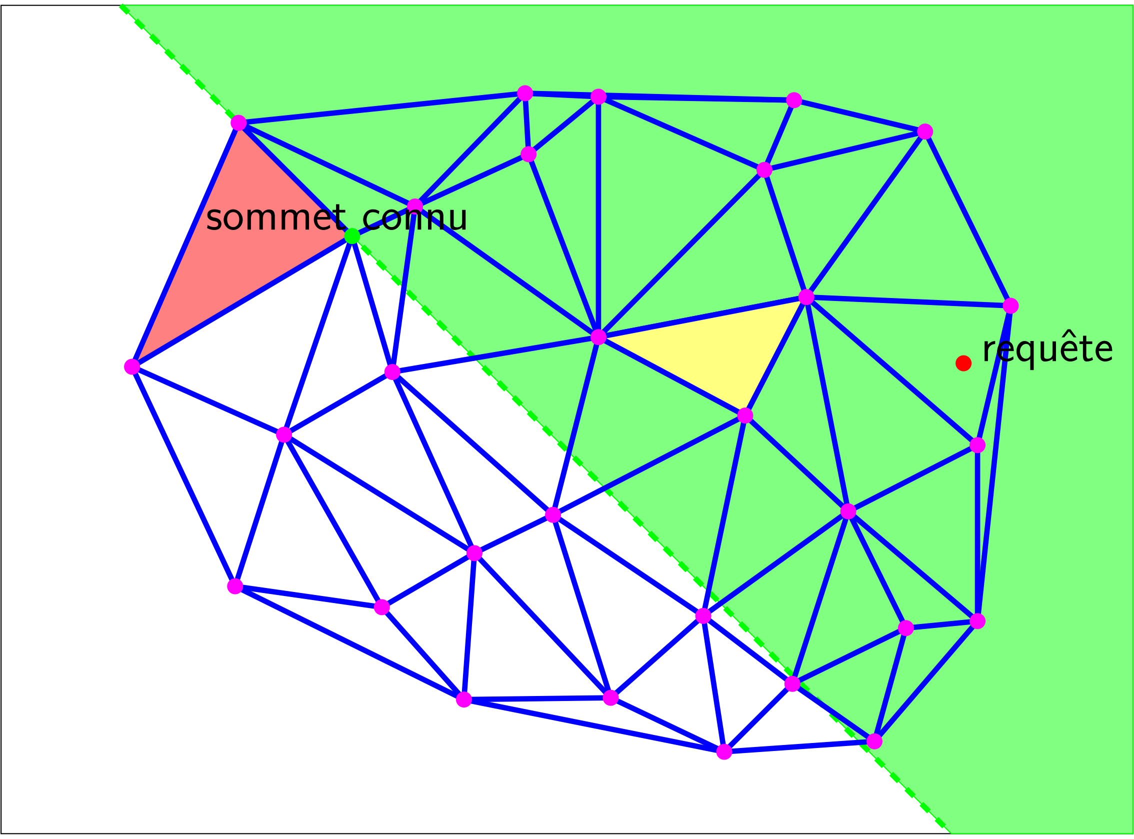


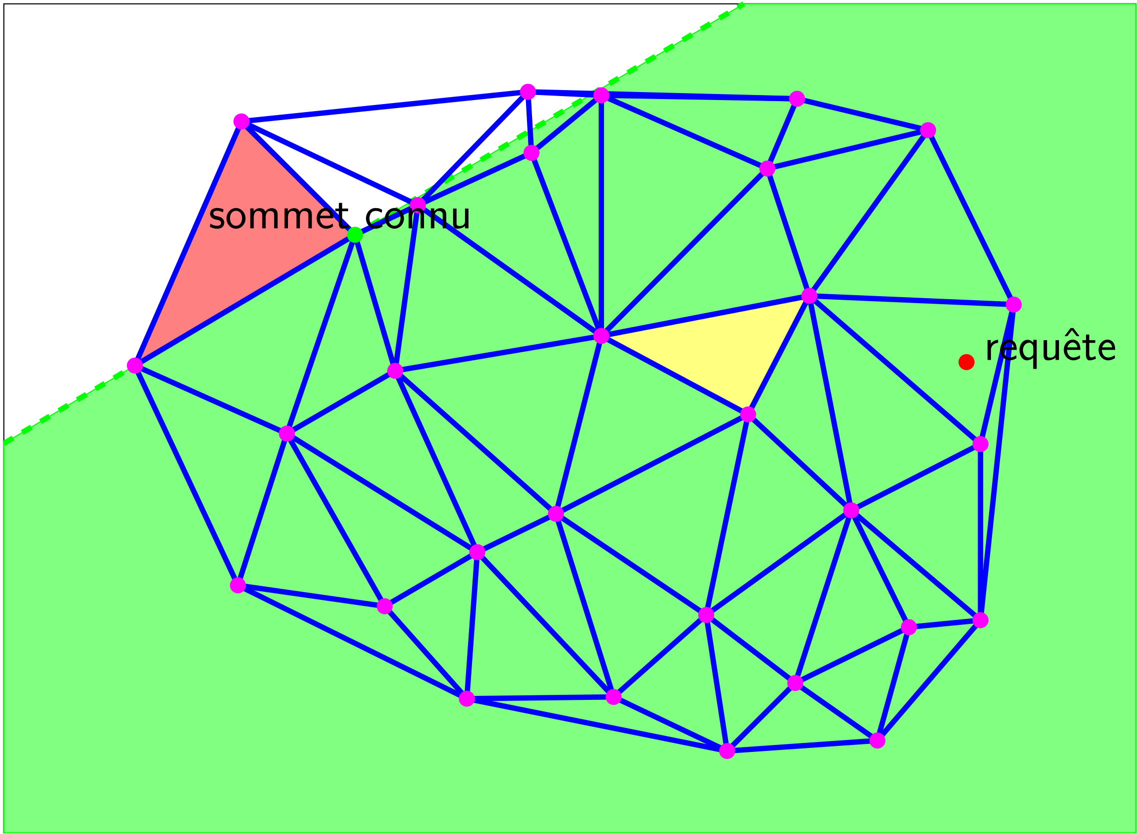
Marcher dans

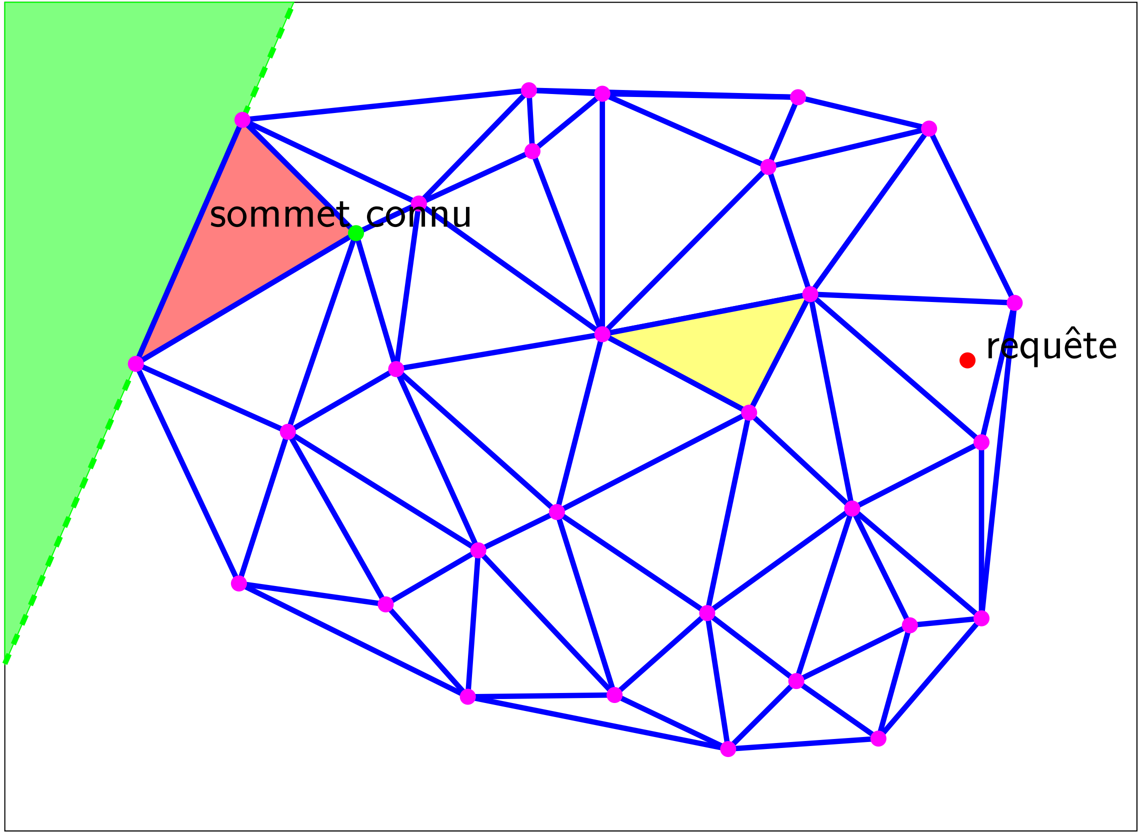
la triangulation de Delaunay

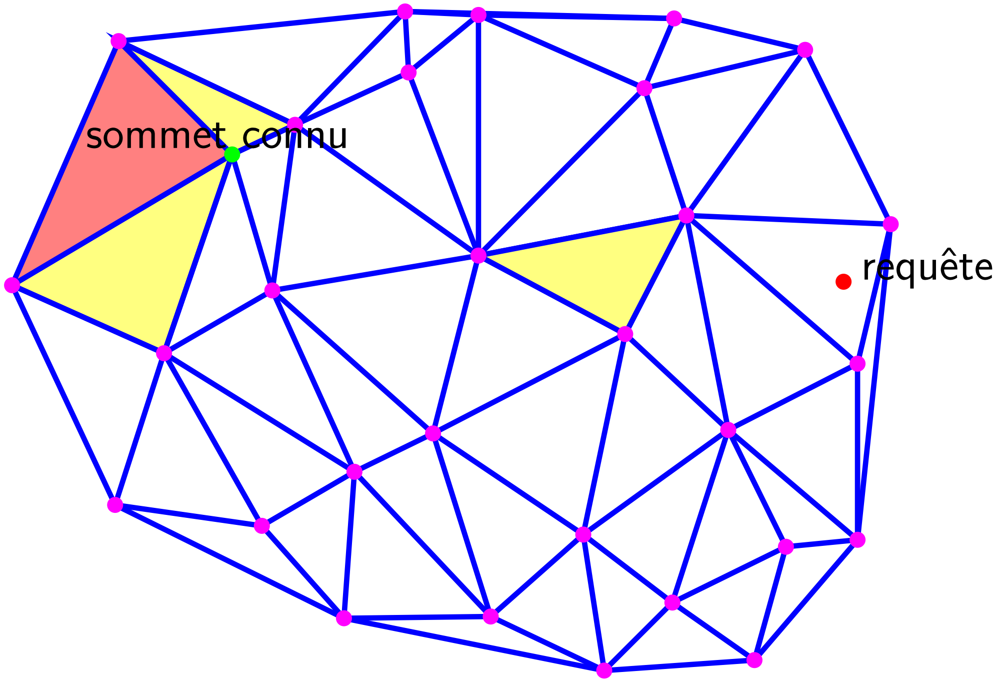
Marche par visibilité

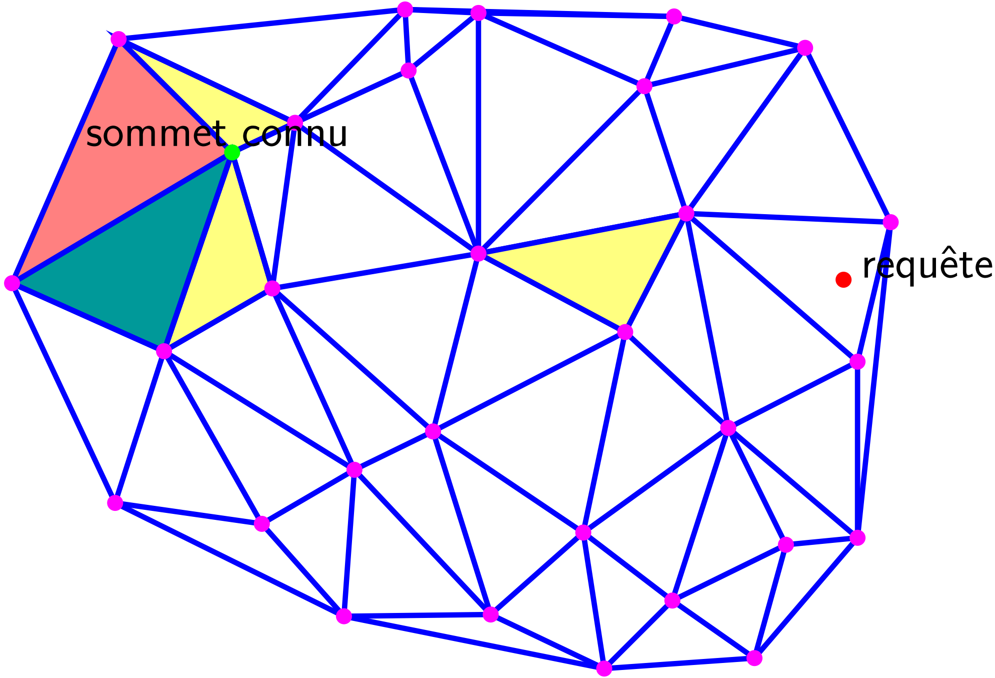

sommet connu
requête

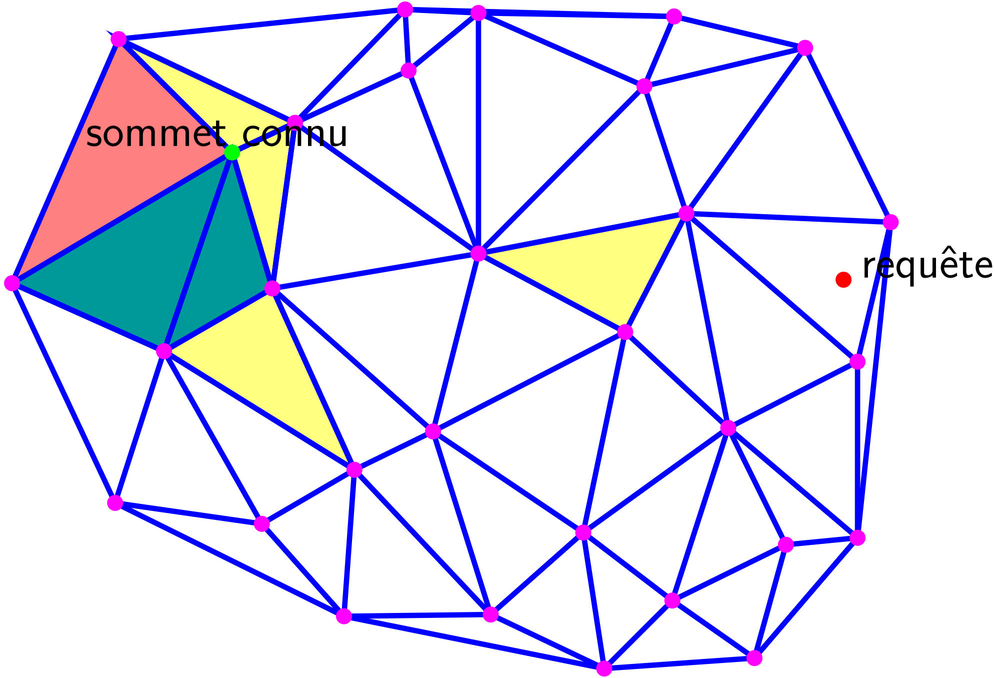

sommet connu
requête

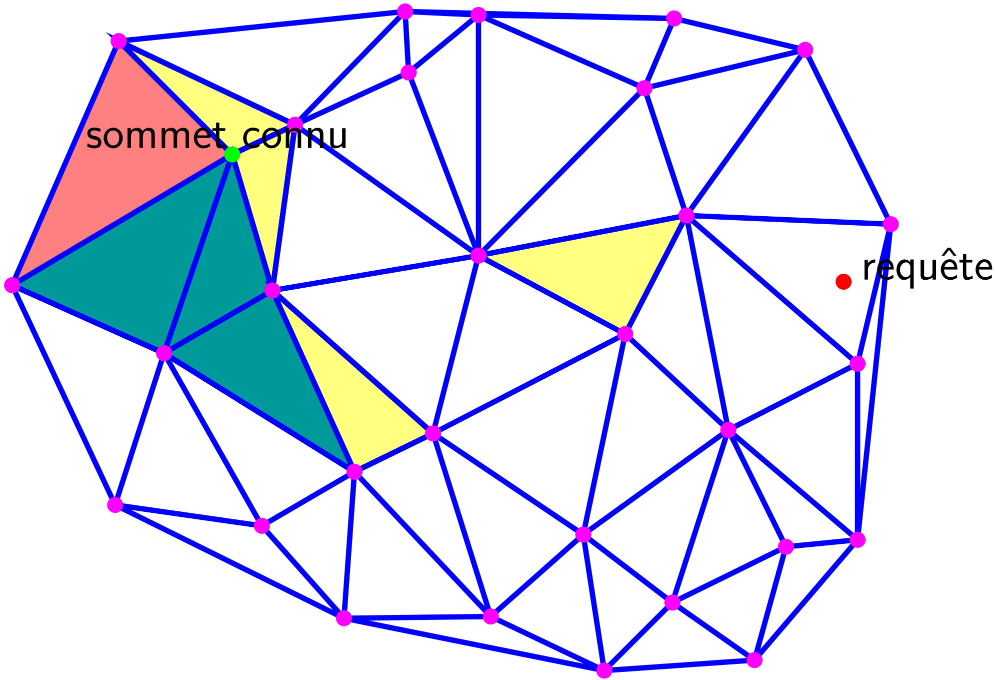

sommet connu
requête

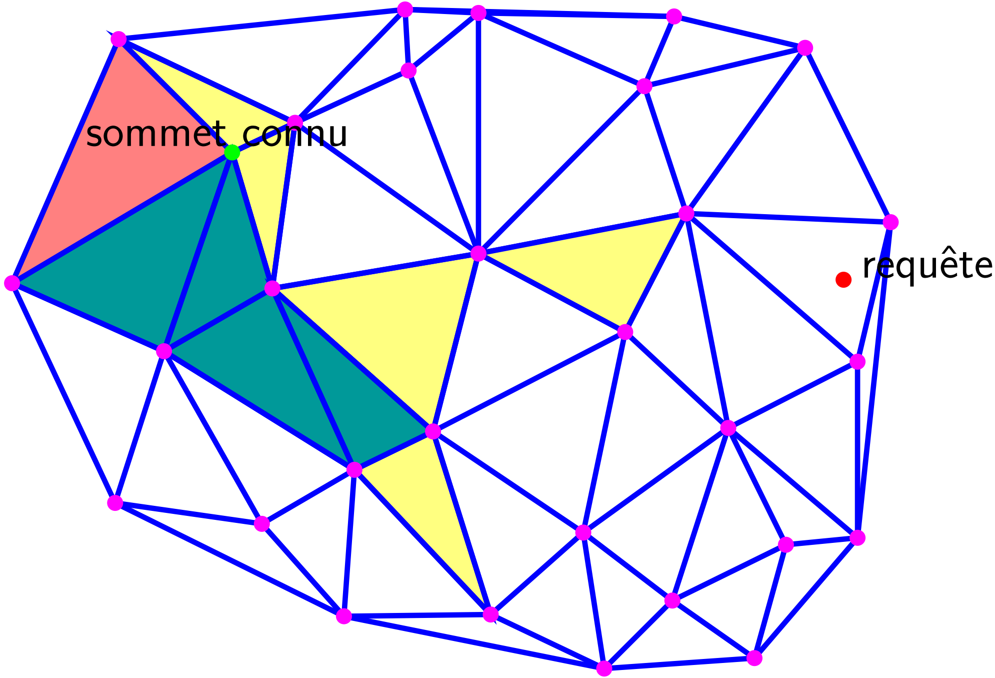

sommet connu
requête

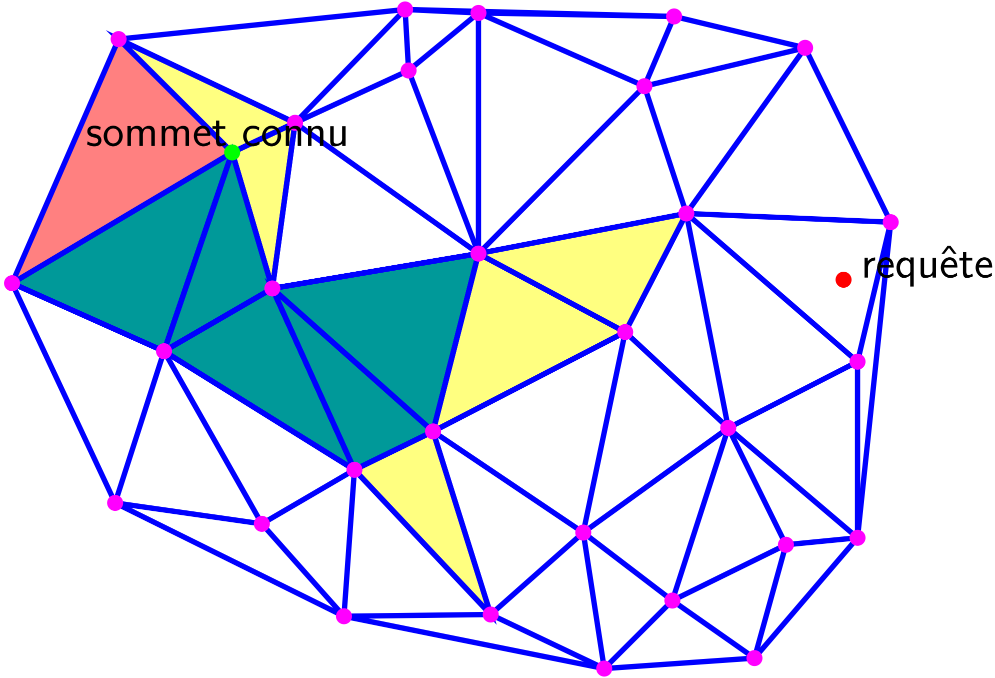

sommet connu
requête

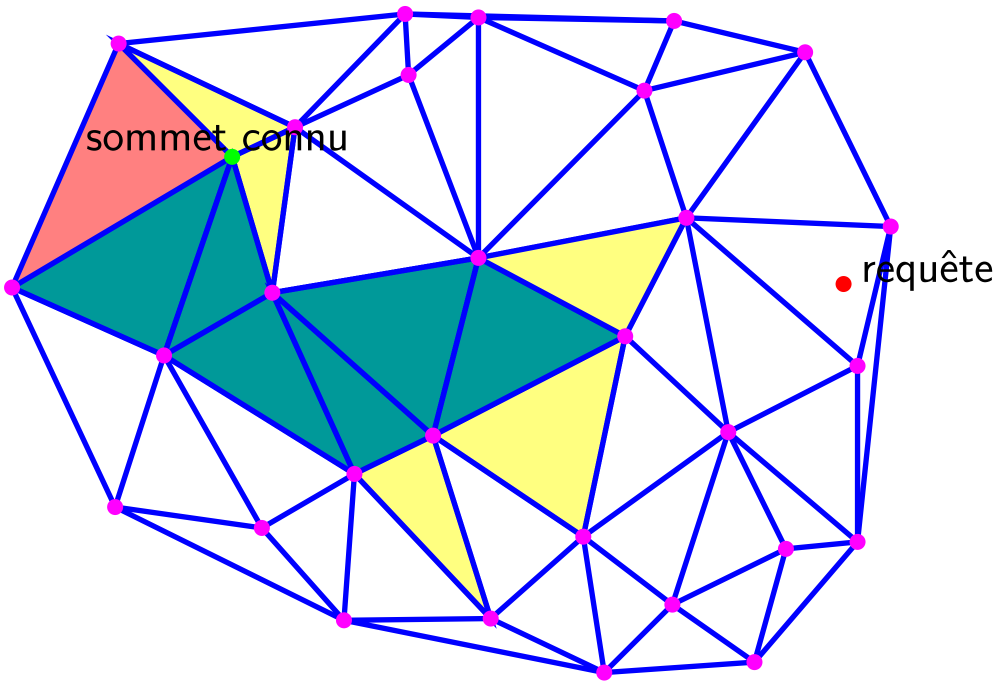

sommet connu
requête

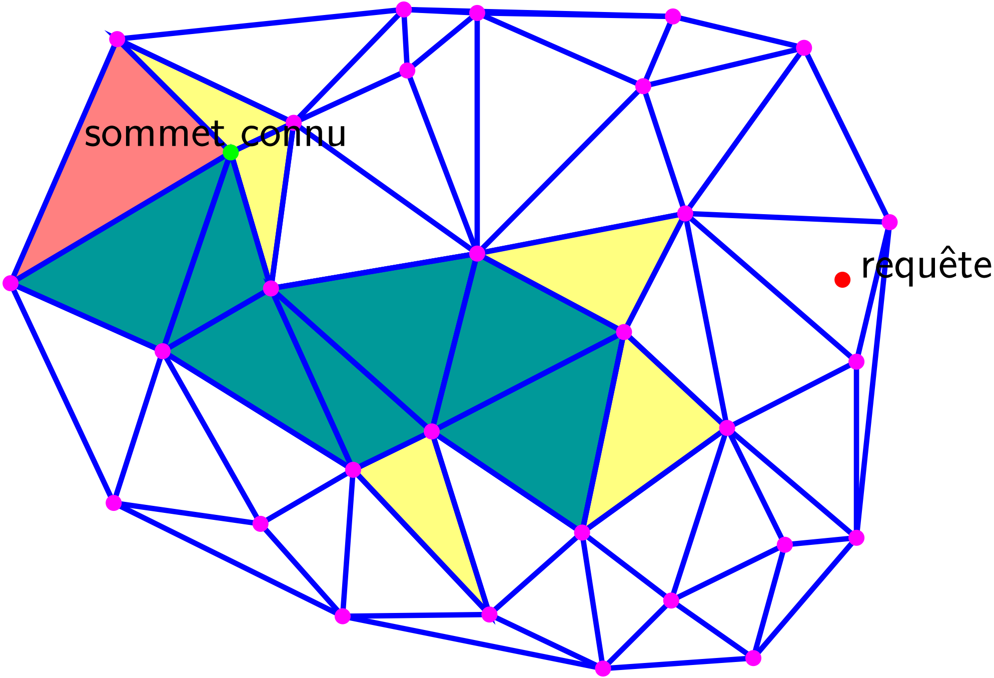

sommet connu
requête

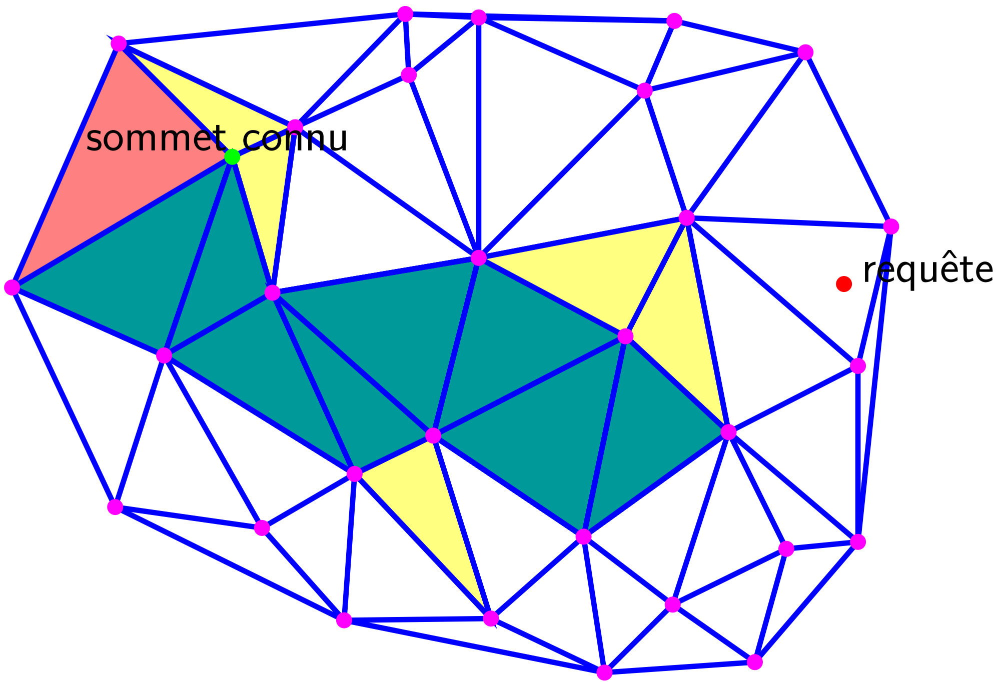

sommet connu
requête


sommet connu
requête

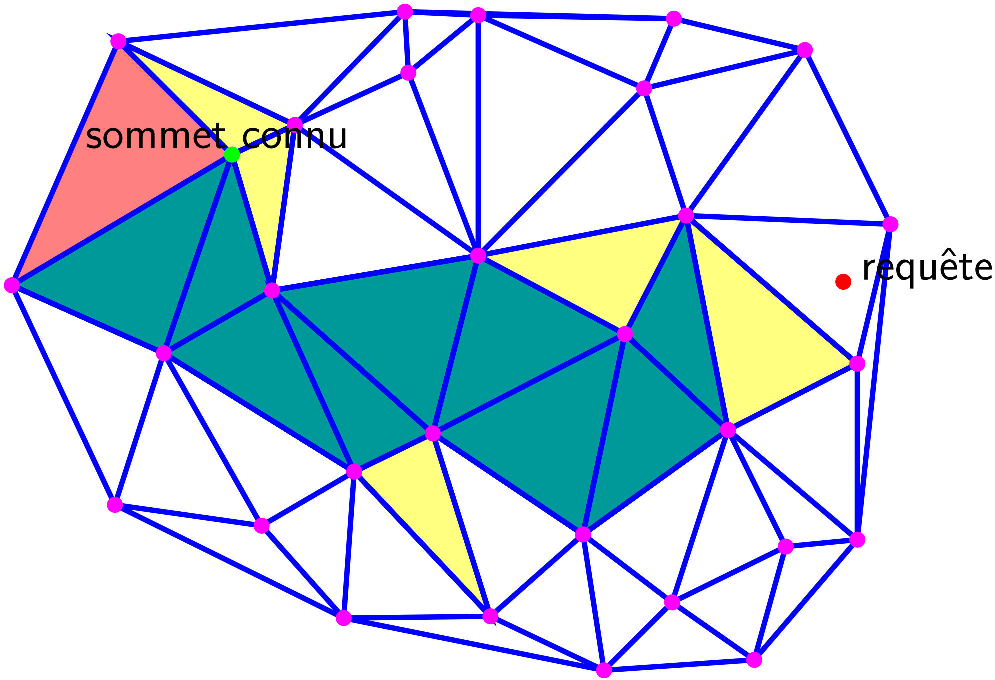

sommet connu
requête

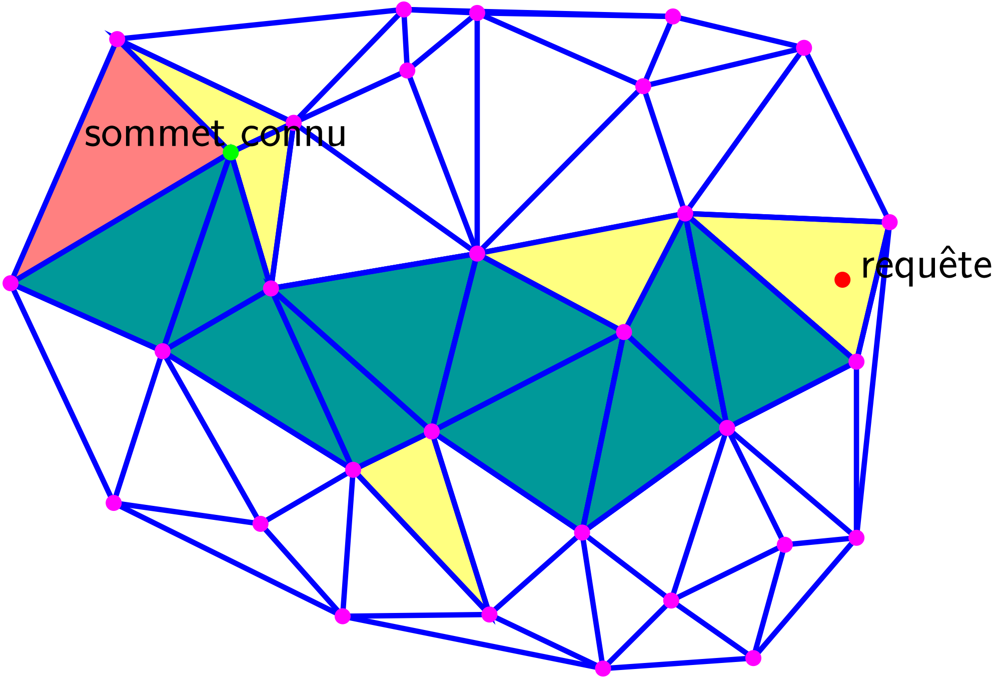

sommet connu
requête

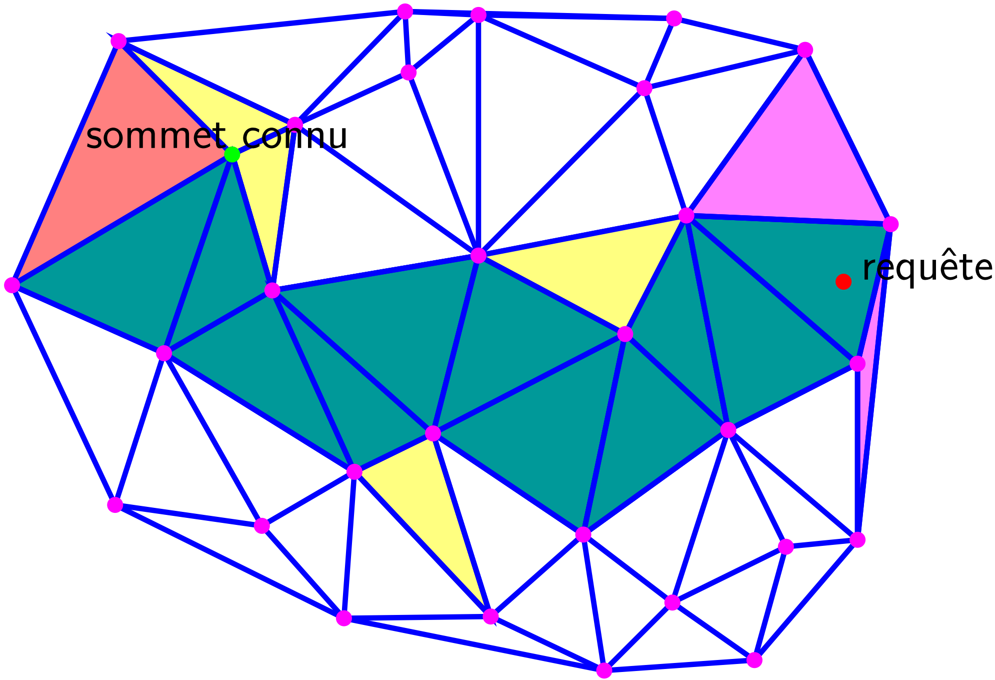

sommet connu
requête



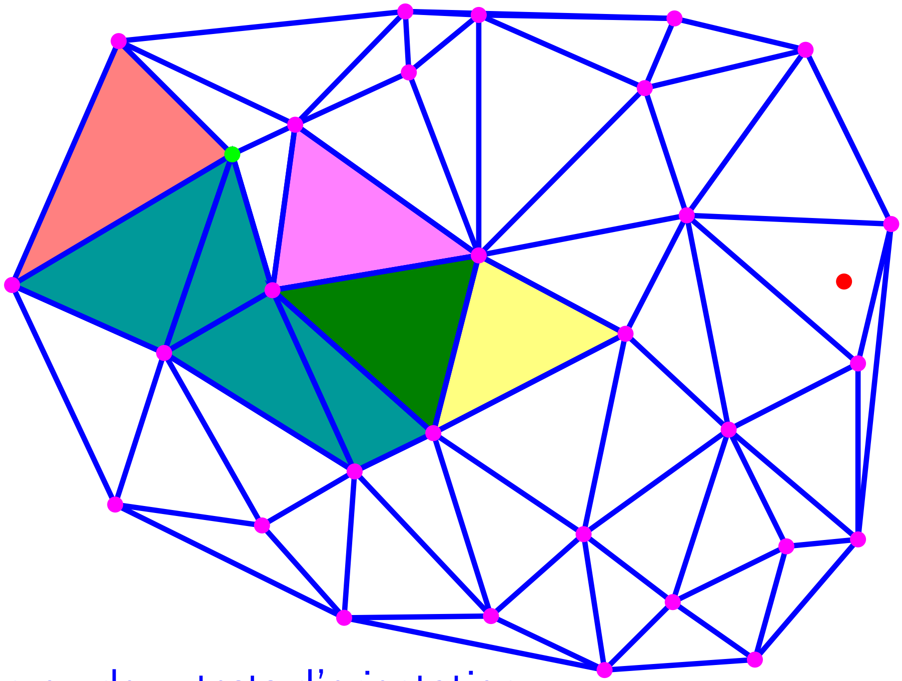
sommet connu
requête


sommet connu
requête

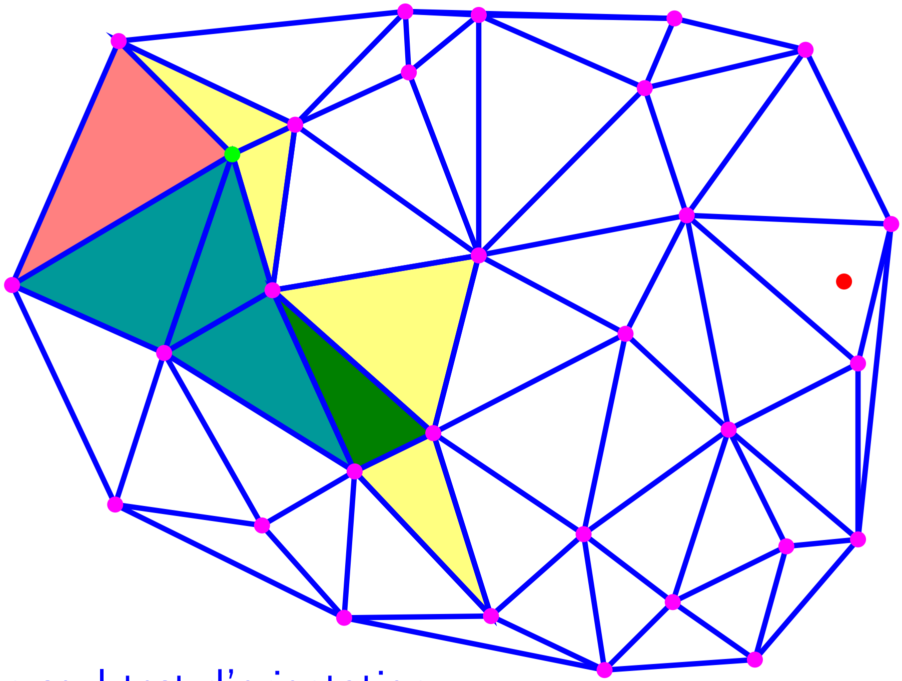

sommet connu
requête



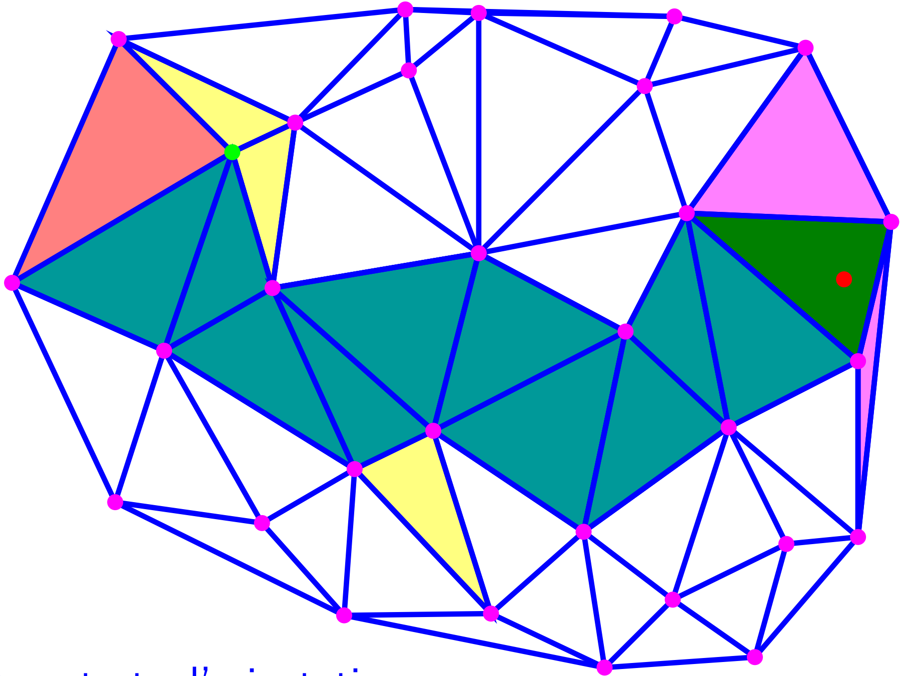
sommet connu
requête



Un ou deux tests d'orientation

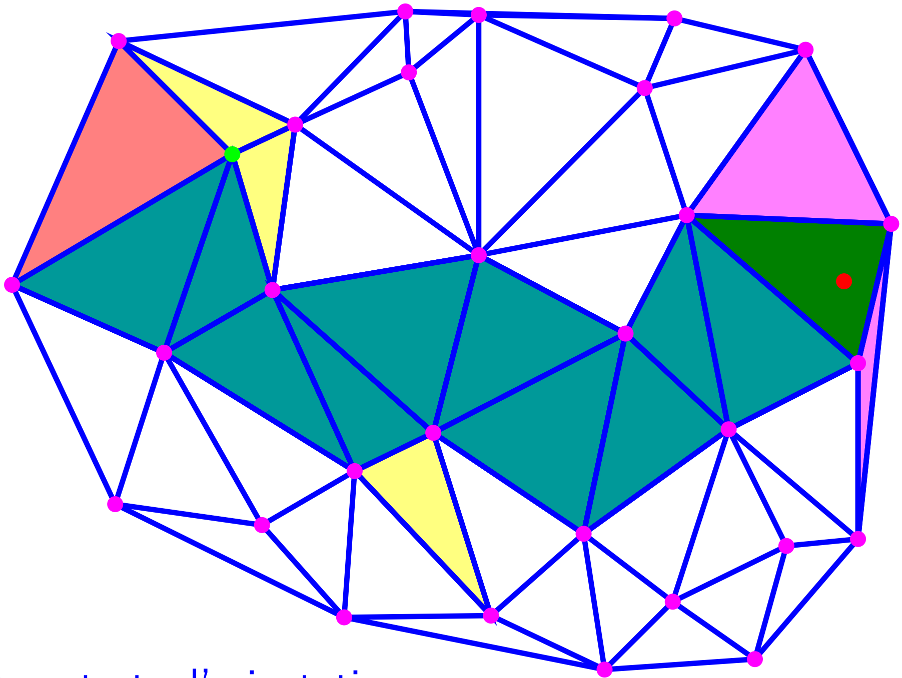


Un seul test d'orientation



Deux tests d'orientation

Combien de triangles ?



Deux tests d'orientation

Combien de triangles ?

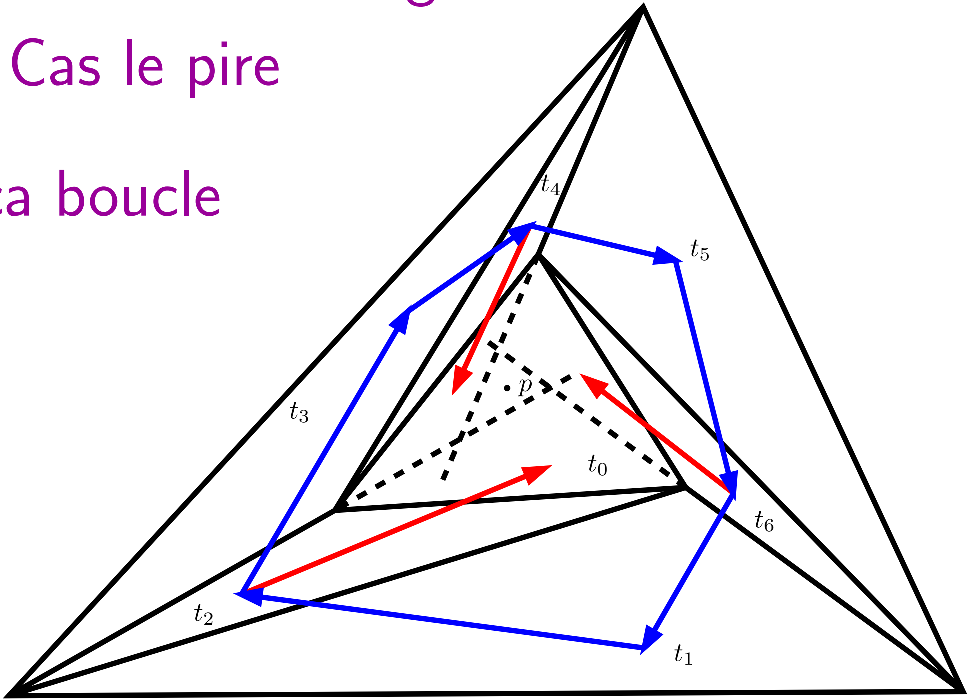
Combien de triangles ?

Cas le pire

Combien de triangles ?

Cas le pire

ça boucle



Combien de triangles ?

Combien de triangles ?

Variante, on choisit au hasard

Combien de triangles ?

Variante, on choisit au hasard

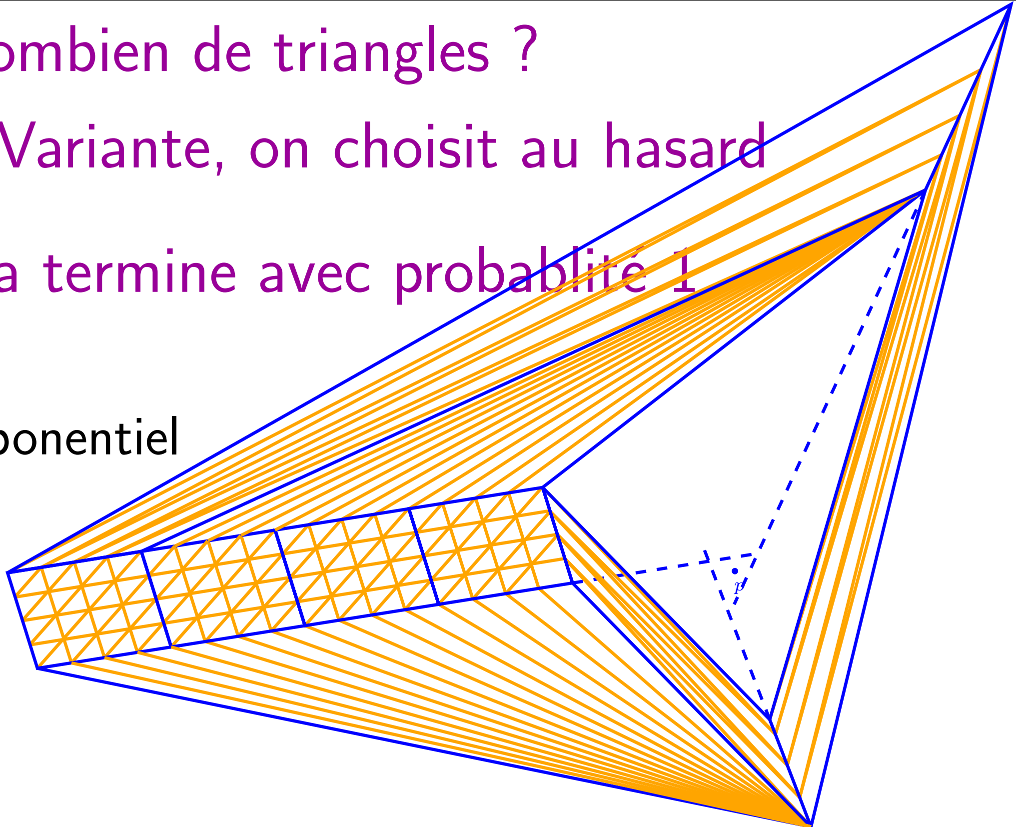
ça termine avec probablité 1

Combien de triangles ?

Variante, on choisit au hasard

ça termine avec probablité 1

exponentiel



Combien de triangles ?

Combien de triangles ?

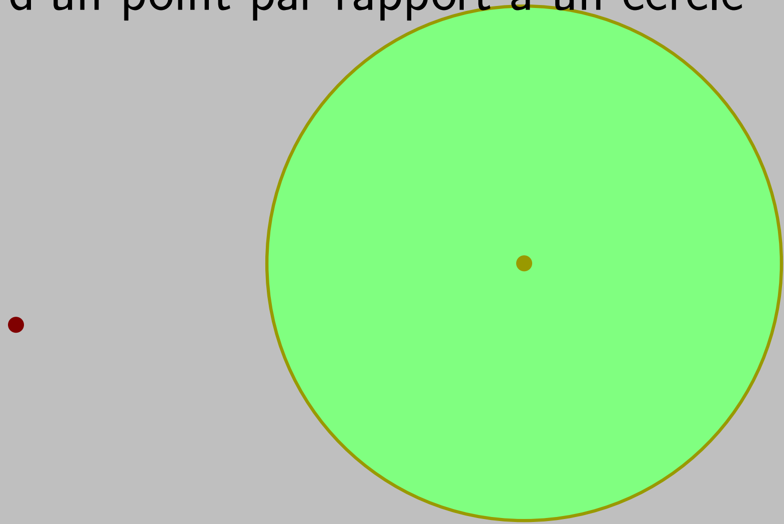
Cas particulier : Delaunay

Combien de triangles ?

Cas particulier : Delaunay

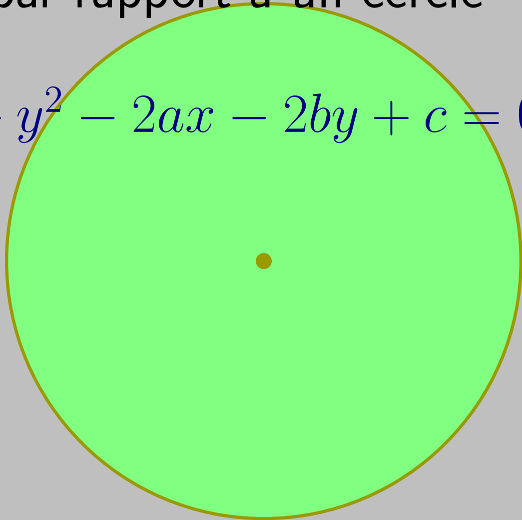
Pas de cycle dans ce cas.

Puissance d'un point par rapport à un cercle



Puissance d'un point par rapport à un cercle

$$x^2 + y^2 - 2ax - 2by + c = 0$$



(x_p, y_p)

avec l'équation

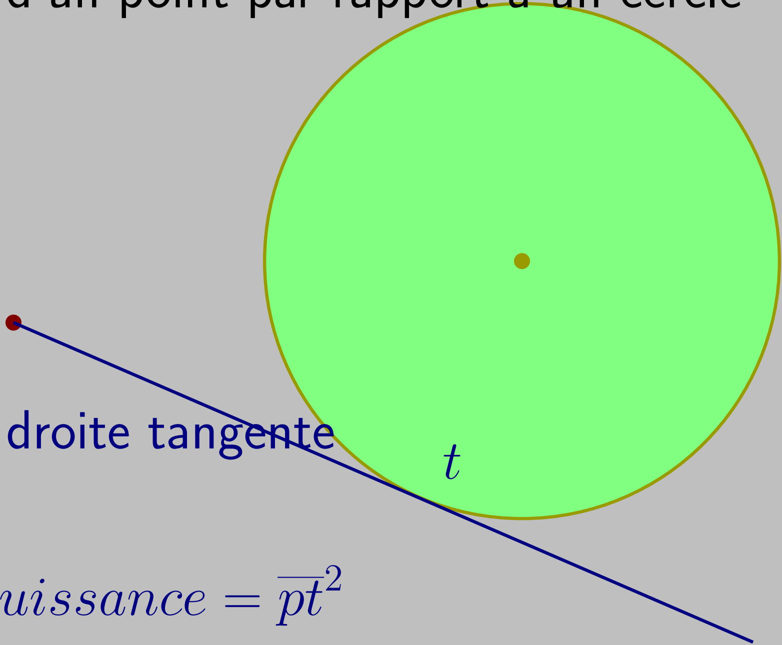
$$puissance = x_p^2 + y_p^2 - 2ax_p - 2by_p + c$$

Puissance d'un point par rapport à un cercle

avec une droite sécante

$$puissance = \overline{pq} \cdot \overline{pr}$$

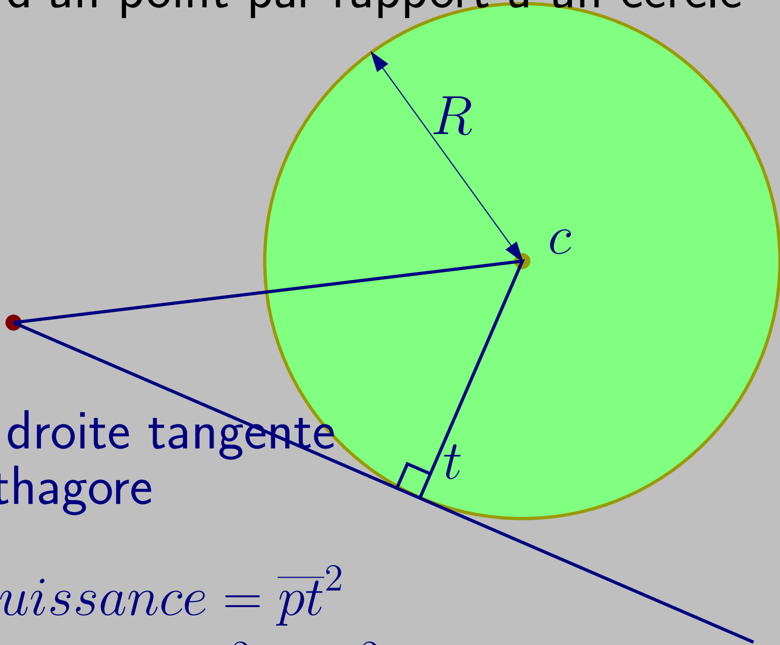
Puissance d'un point par rapport à un cercle



avec une droite tangente

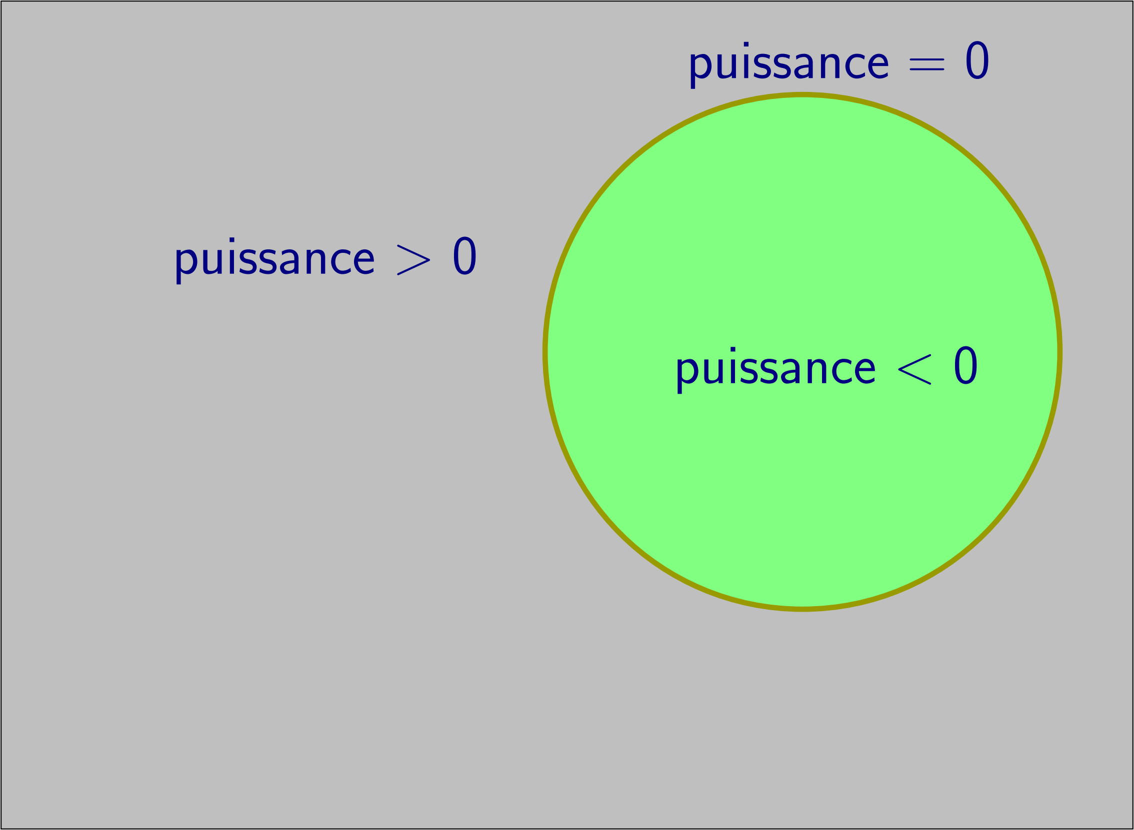
$$puissance = \overline{pt}^2$$

Puissance d'un point par rapport à un cercle

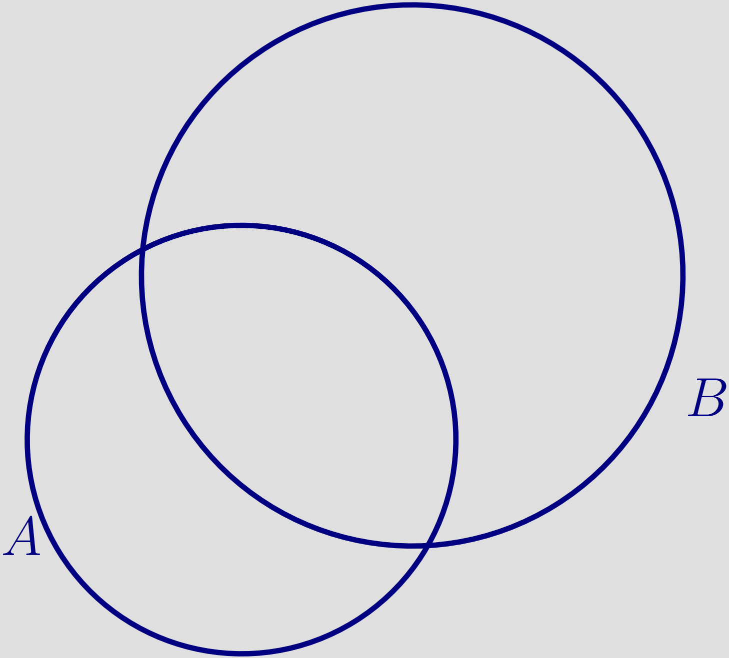


avec une droite tangente
et Pythagore

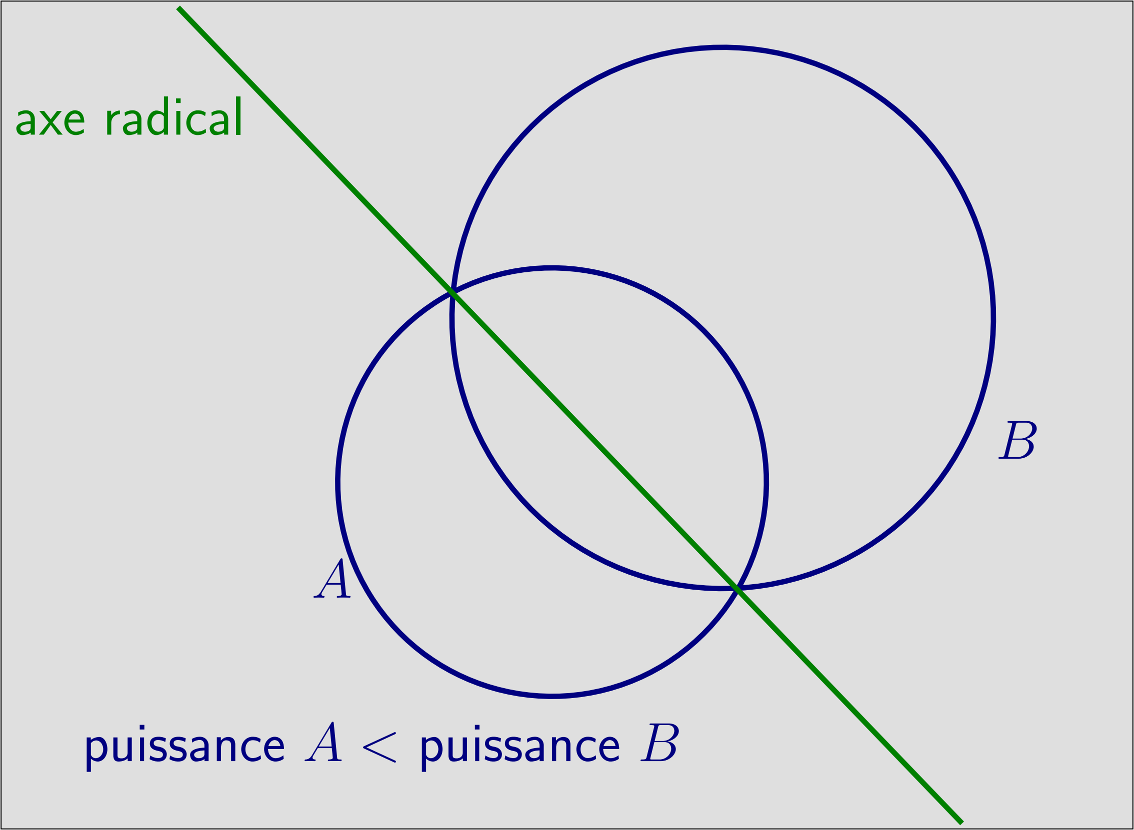
$$puissance = \overline{pt}^2$$
$$= \overline{pc}^2 - R^2$$

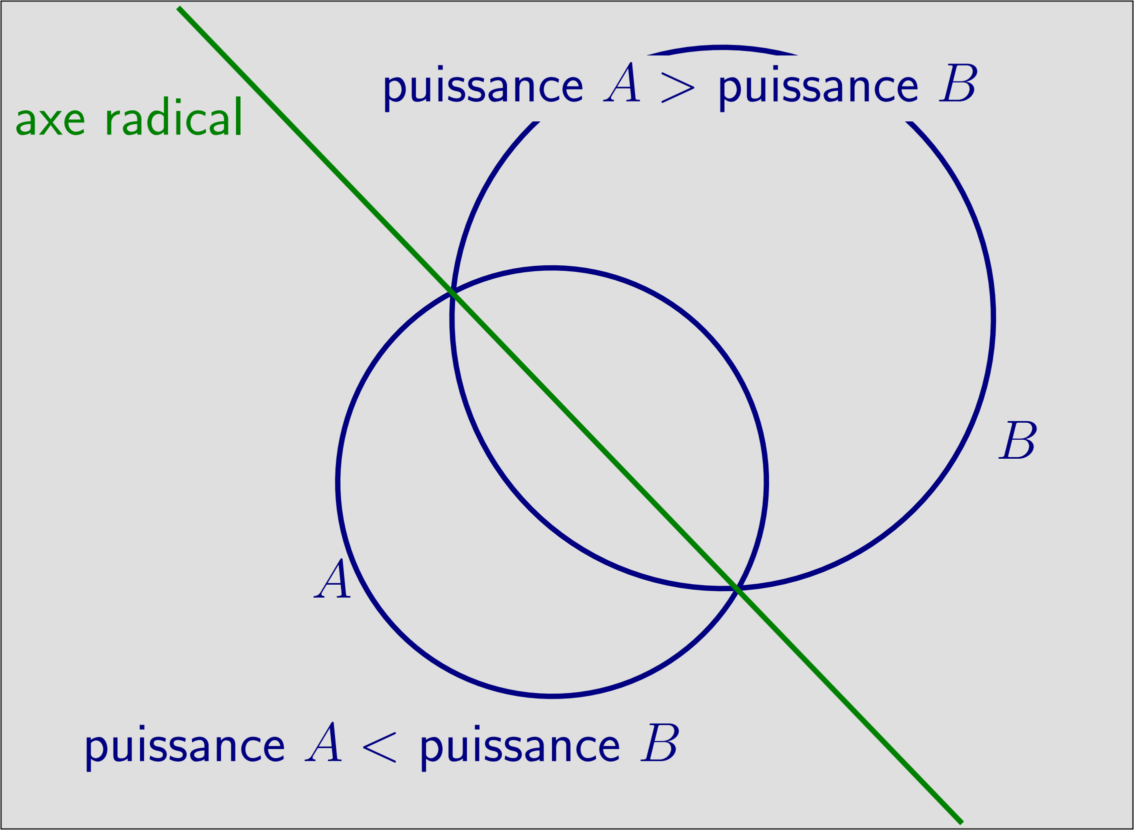


puissance = 0
puissance > 0
puissance < 0



puissance $A <$ puissance B


axe radical
B
A
puissance $A <$ puissance B


axe radical
puissance $A >$ puissance B
B
A
puissance $A <$ puissance B

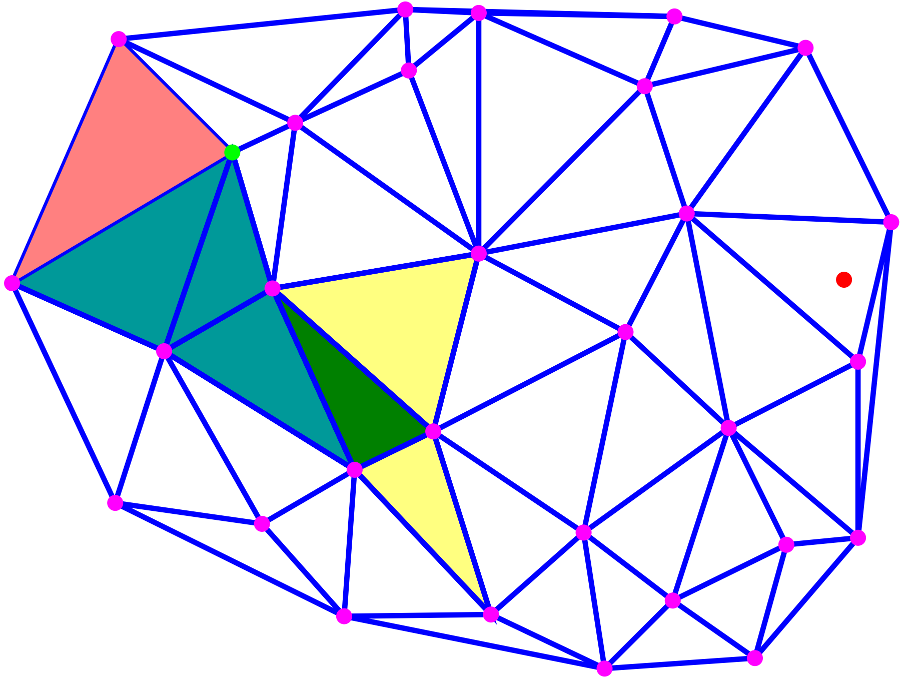
Combien de triangles ?

Cas particulier : Delaunay

Pas de cycle dans ce cas.

Combien de triangles ?

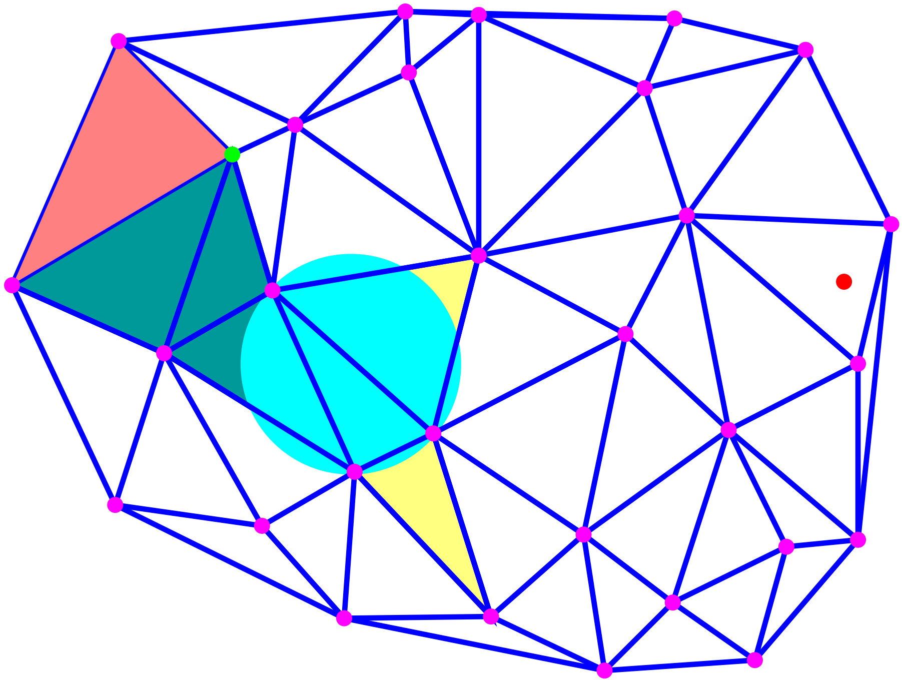
Cas particulier : Delaunay



Pas de cycle dans ce cas.

Combien de triangles ?

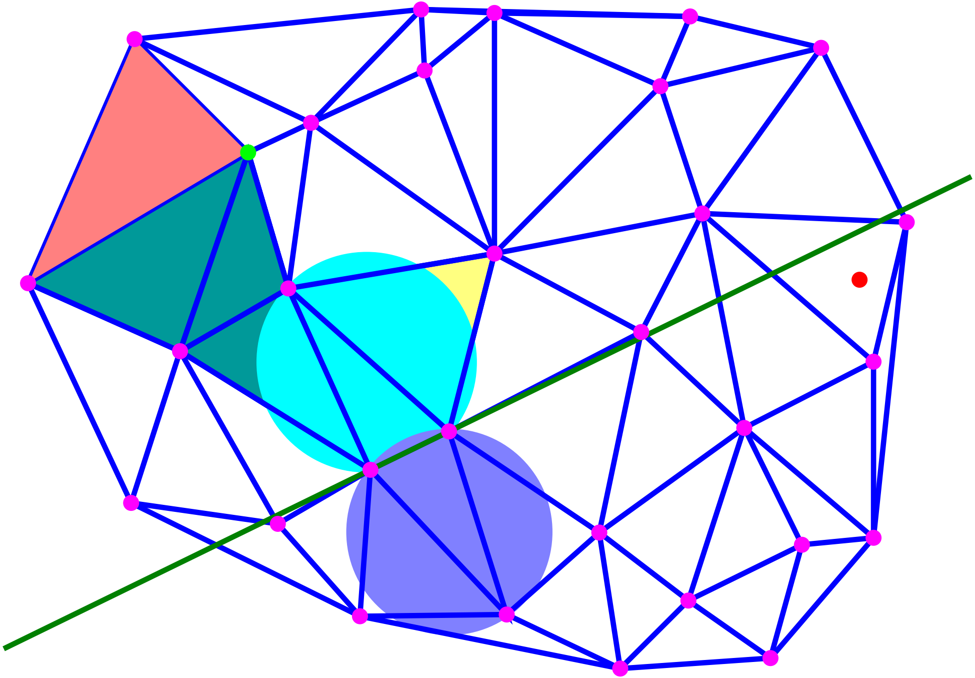
Cas particulier : Delaunay



Pas de cycle dans ce cas.

Combien de triangles ?

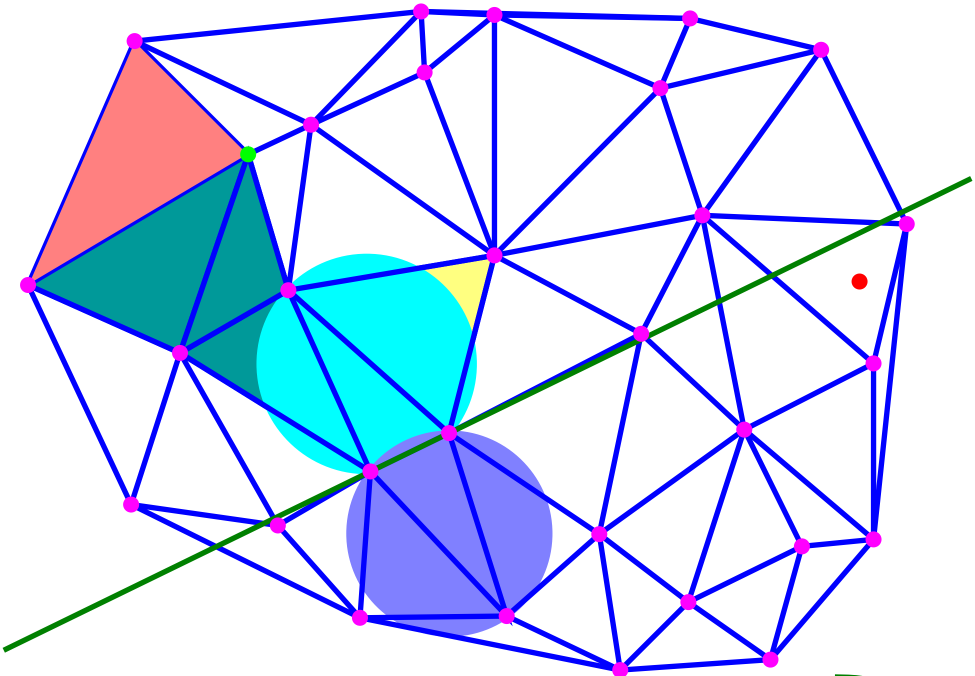
Cas particulier : Delaunay



Pas de cycle dans ce cas.

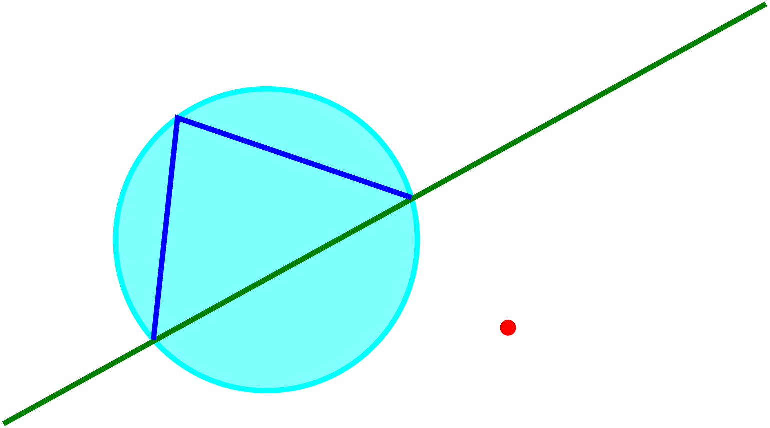
Combien de triangles ?

Cas particulier : Delaunay

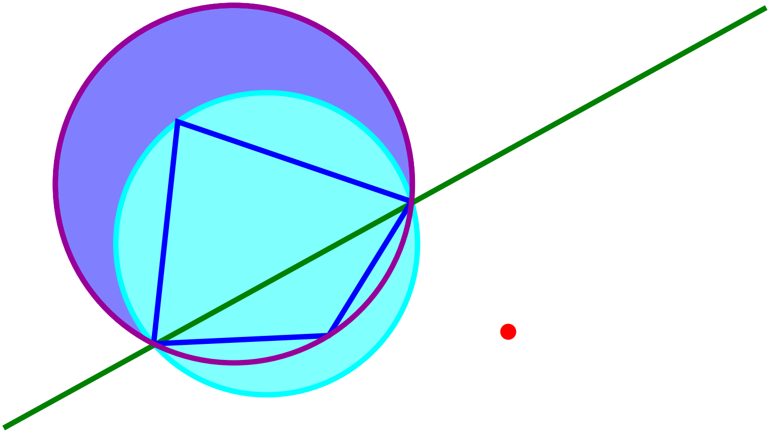


Pas de cycle dans ce cas. car puissance(•)

Si c'est pas Delaunay



Si c'est pas Delaunay



La puissance peut augmenter

Combien de triangles ?

Cas particulier : Delaunay

Pas de cycle dans ce cas.

$2n$ triangles maximum

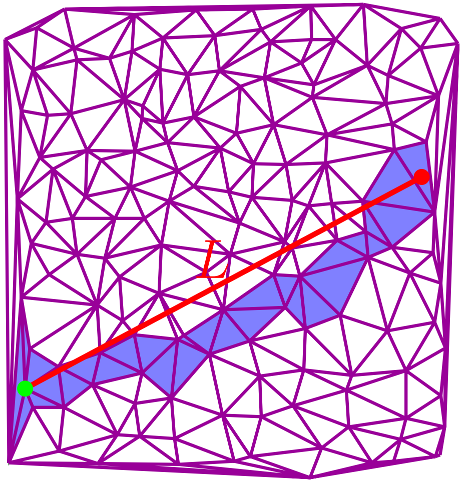
Combien de triangles ?

Points uniformément distribués

Combien de triangles ?

Points uniformément distribués

Même intuition



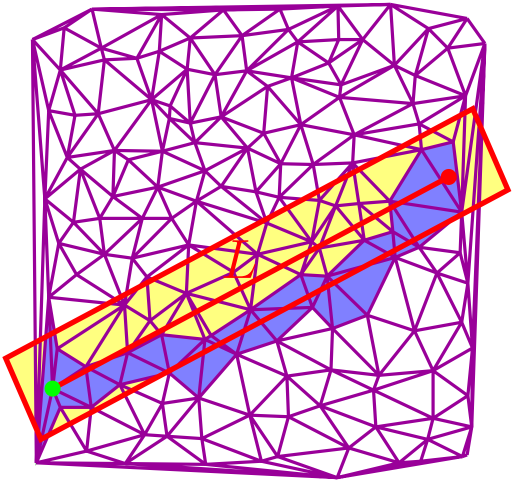
Combien de triangles ?

Points uniformément distribués

Même intuition

Nombre de triangles visités

$$\simeq 2L\sqrt{n}$$



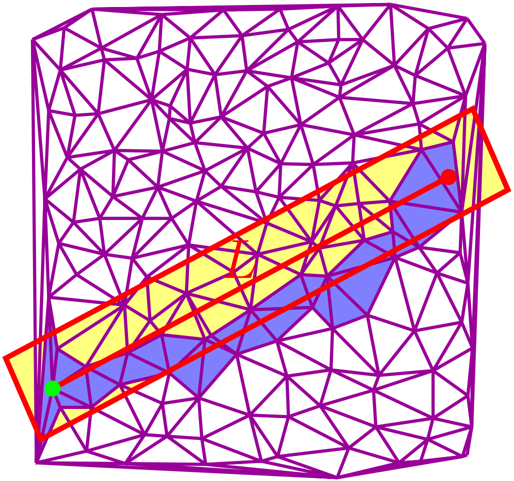
Combien de triangles ?

Points uniformément distribués

Même intuition

Nombre de triangles visités

$$\simeq 2L\sqrt{n}$$



Preuve théorique

Problème ouvert !

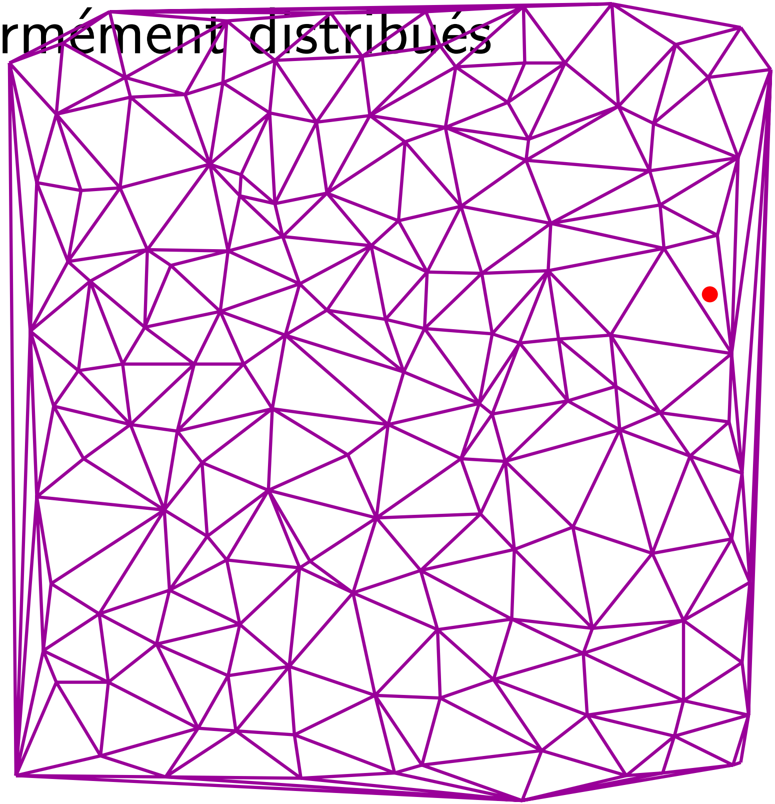
Analyse randomisée pour

la triangulation de Delaunay

la localisation

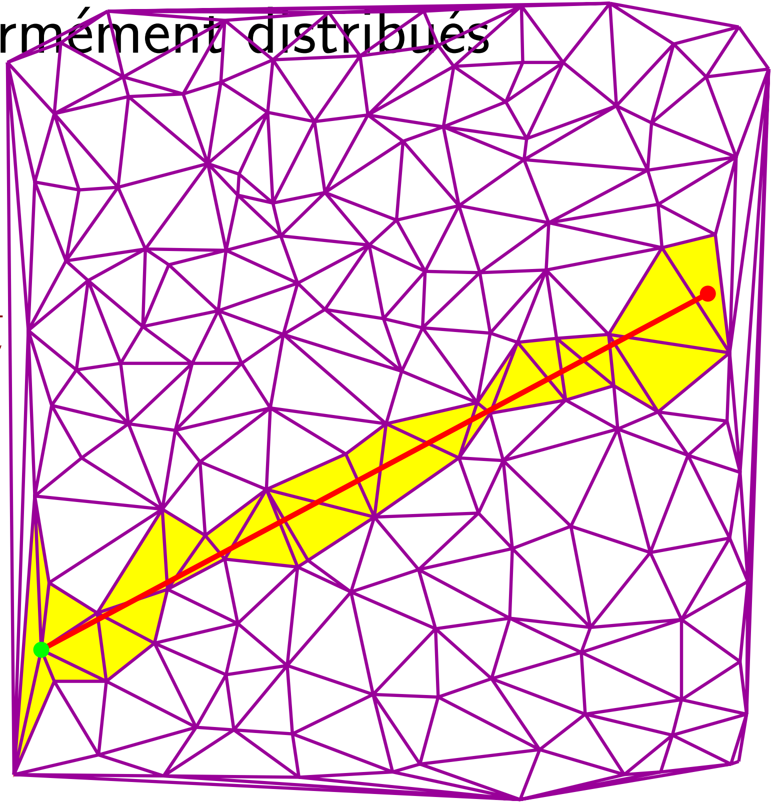
Jump and walk

Points uniformément distribués



Points uniformément distribués

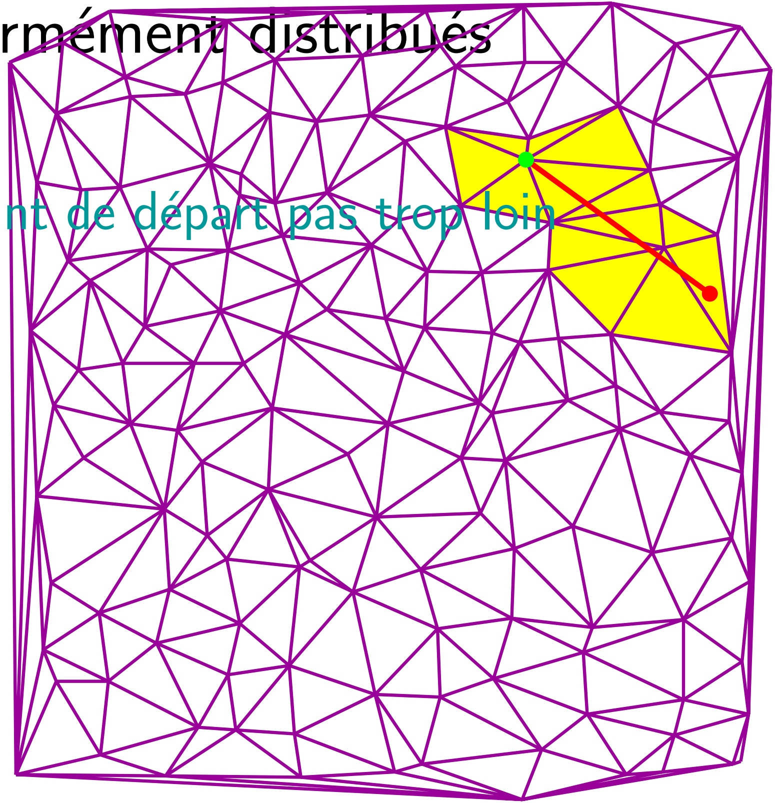
$$2L\sqrt{n} \leq 2\sqrt{2}\sqrt{n}$$



Points uniformément distribués

Choisir un point de départ pas trop loin

$2L\sqrt{n}\,;\ L << 1$

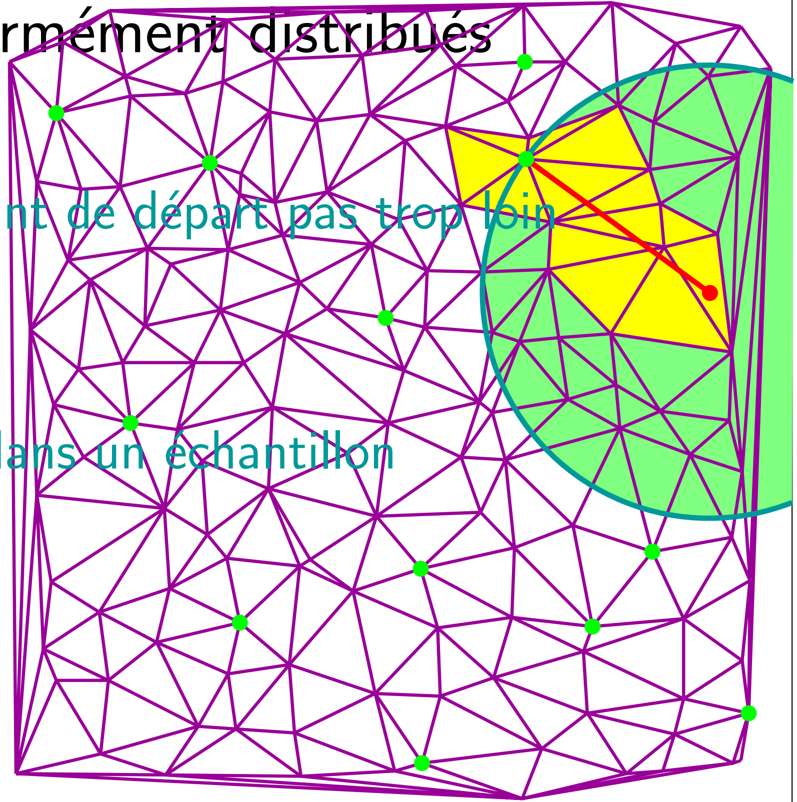


Points uniformément distribués

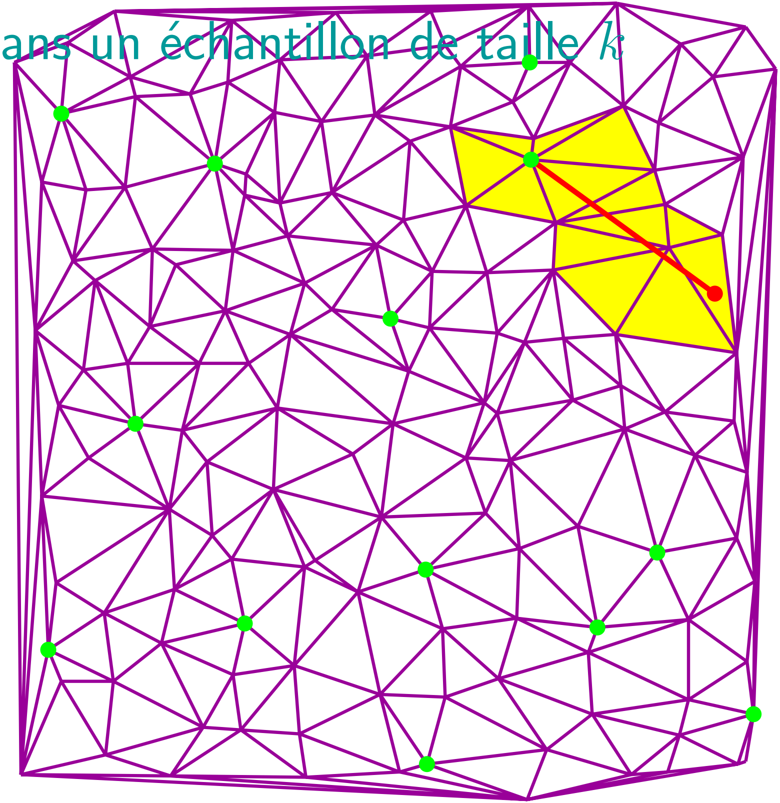
Choisir un point de départ pas trop loin

$2L\sqrt{n}\ ;\ L << 1$

Le plus près dans un échantillon



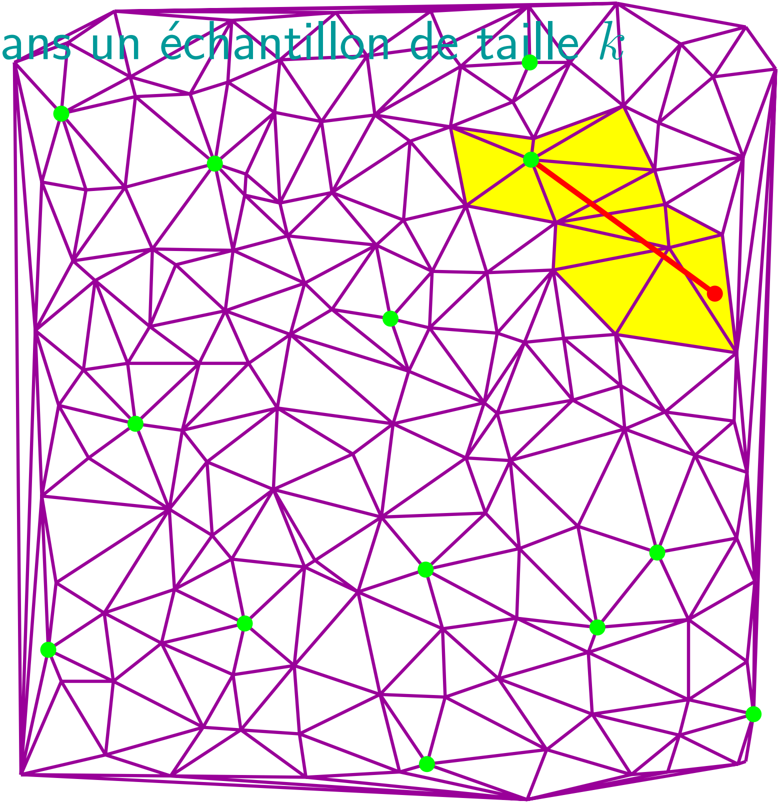
Le plus près dans un échantillon de taille k



Le plus près dans un échantillon de taille k

$$L \simeq \frac{1}{\sqrt{k}}$$

$$L\sqrt{n} \simeq \frac{\sqrt{n}}{\sqrt{k}}$$

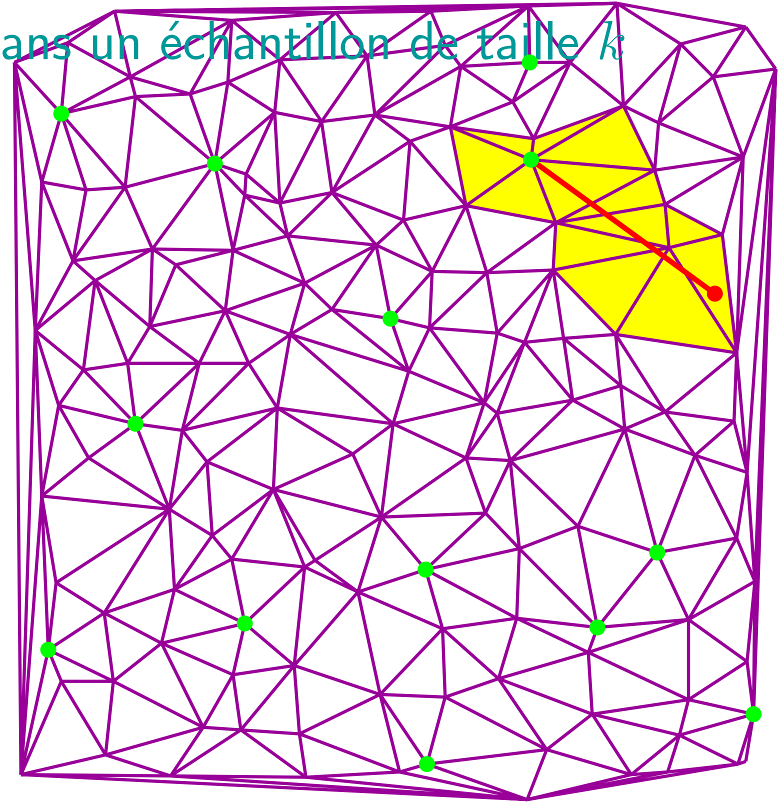


Le plus près dans un échantillon de taille k

$$L \simeq \frac{1}{\sqrt{k}}$$

$$L\sqrt{n} \simeq \frac{\sqrt{n}}{\sqrt{k}}$$

$$k + \sqrt{\frac{n}{k}}$$



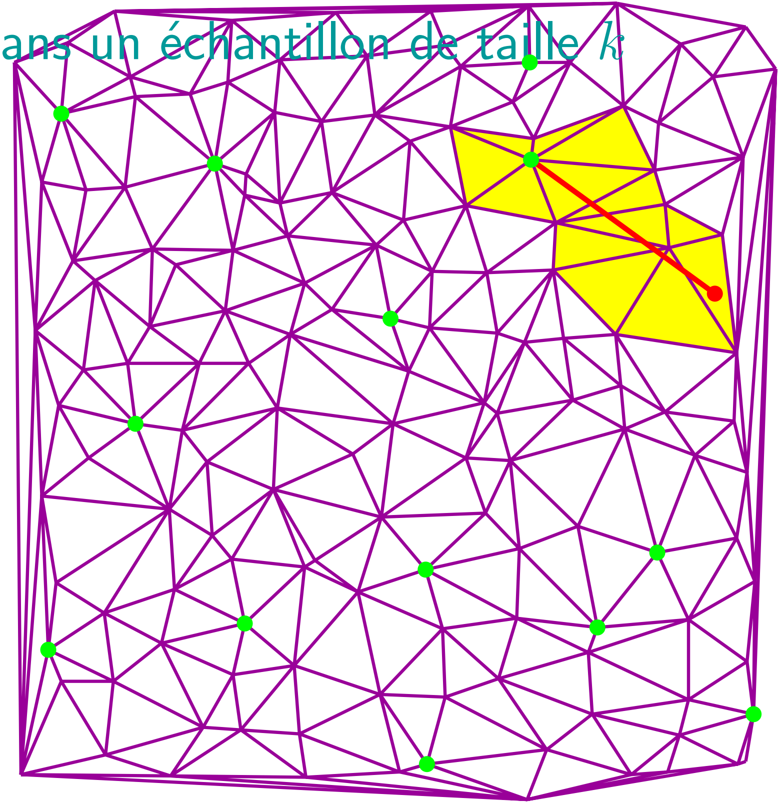
Le plus près dans un échantillon de taille k

$$L \simeq \frac{1}{\sqrt{k}}$$

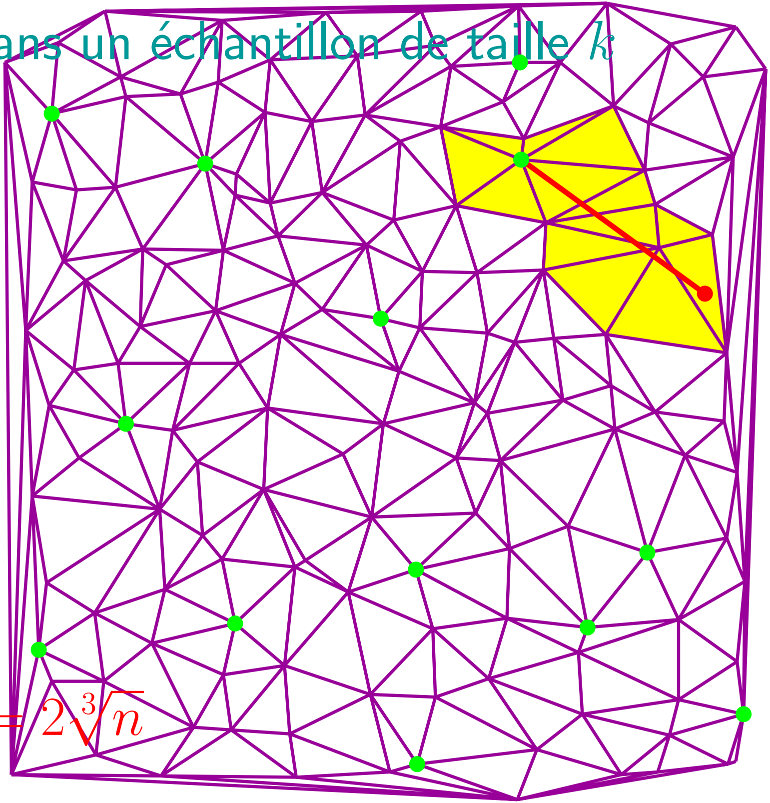
$$L\sqrt{n} \simeq \frac{\sqrt{n}}{\sqrt{k}}$$

$$k + \sqrt{\frac{n}{k}}$$

$$k = \sqrt[3]{n}$$



Le plus près dans un échantillon de taille k



$$L \simeq \frac{1}{\sqrt{k}}$$

$$L\sqrt{n} \simeq \frac{\sqrt{n}}{\sqrt{k}}$$

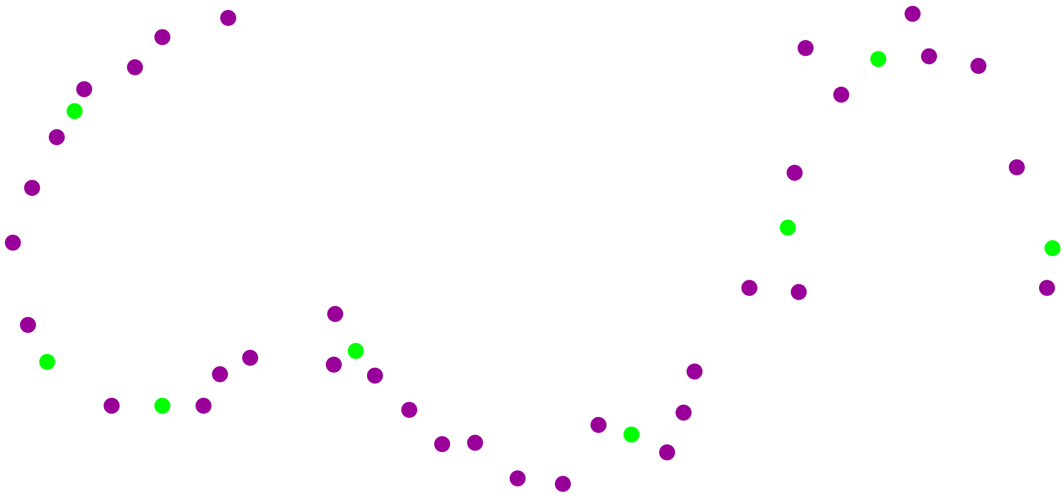
$$k + \sqrt{\frac{n}{k}}$$

$$k = \sqrt[3]{n}$$

$$\sqrt[3]{n} + \sqrt{\frac{n}{\sqrt[3]{n}+}} = 2\sqrt[3]{n}$$

Si points NON uniformément distribués

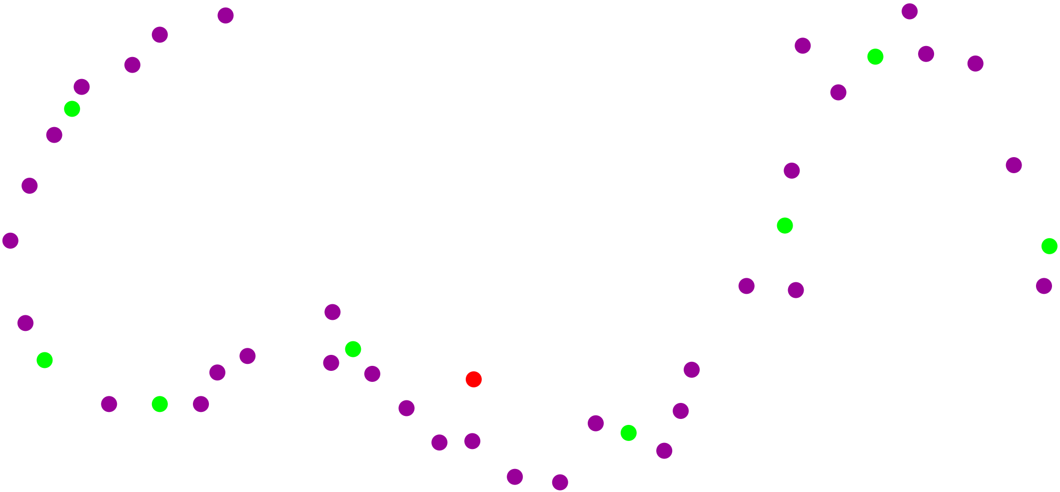
Le plus près dans un échantillon de taille k



Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

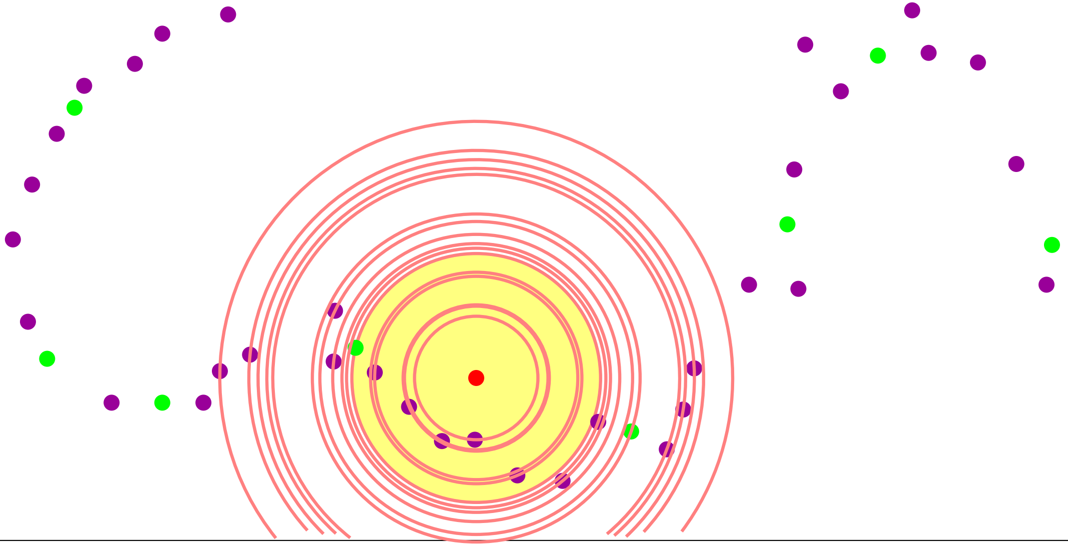


Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête



Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête

Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête

Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête

Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête

$\frac{n}{k}$

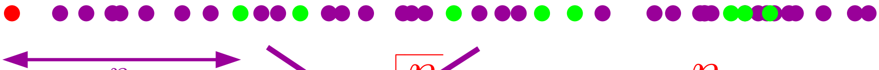
$$k + \sqrt{\frac{n}{k}}$$

Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête



$\frac{n}{k}$

$\cancel{k+\sqrt{\frac{n}{k}}}$

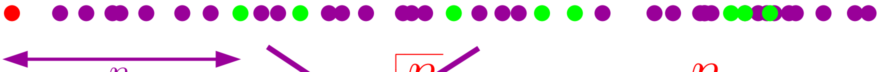
$k+\frac{n}{k}$

Si points NON uniformément distribués

Le plus près dans un échantillon de taille k

requête

seul compte l'ordre dans la distance à la requête



$$\frac{n}{k}$$

$$\cancel{k+\sqrt{\frac{n}{k}}} \qquad k+\frac{n}{k}$$

$$k=\sqrt[2]{n} \qquad \sqrt{n}+\frac{n}{\sqrt{n}}=2\sqrt{n}$$

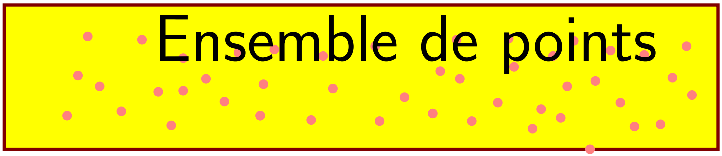
Analyse randomisée pour

la triangulation de Delaunay

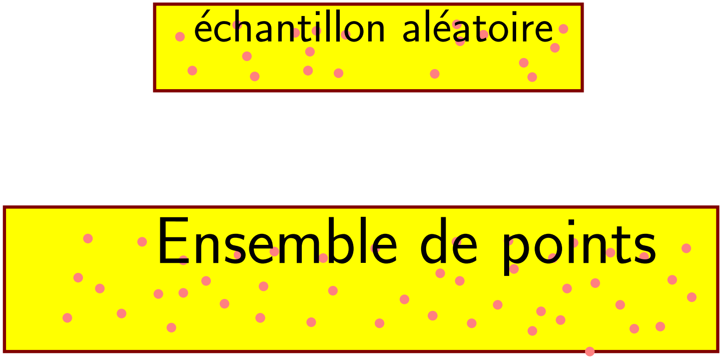
la localisation

Hiérarchie de Delaunay

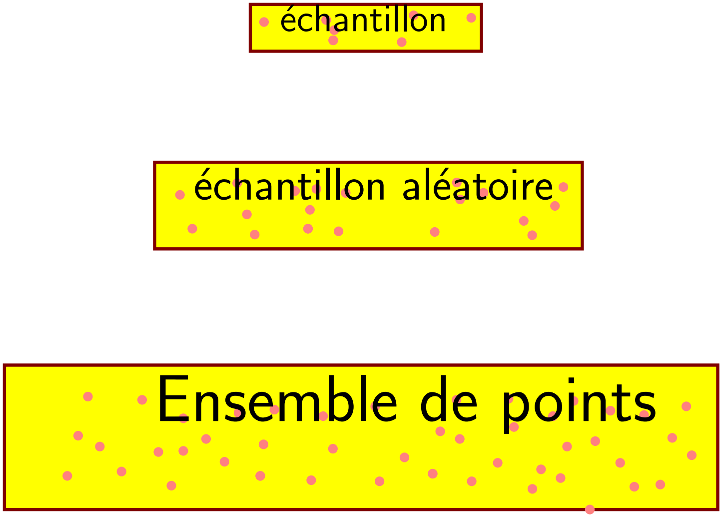
Hiérarchie de Delaunay



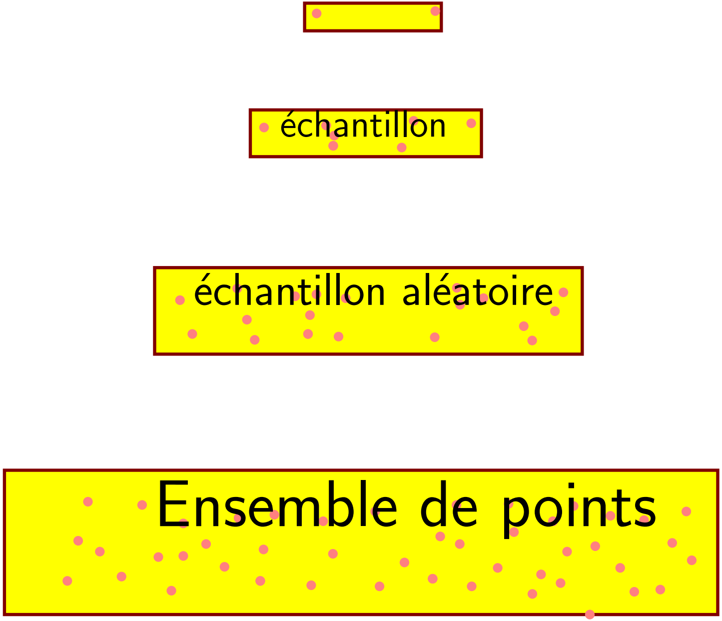
Hiérarchie de Delaunay



Hiérarchie de Delaunay

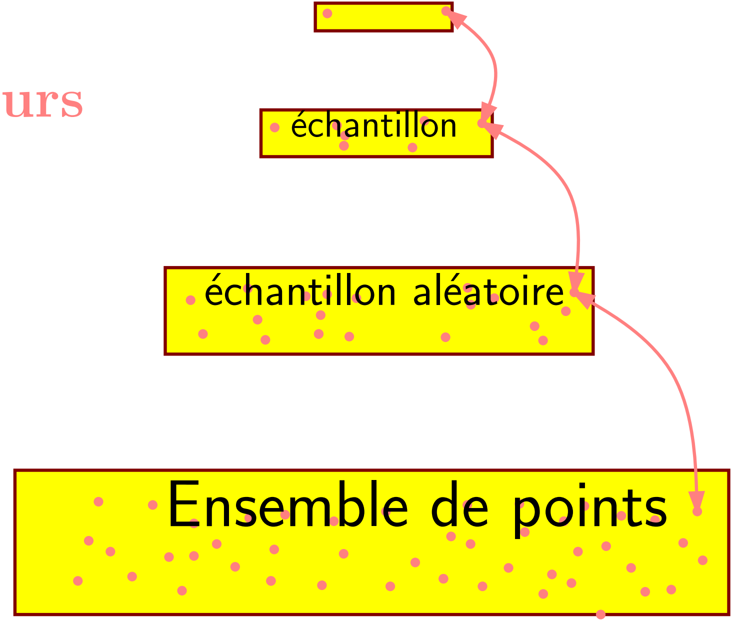


Hiérarchie de Delaunay



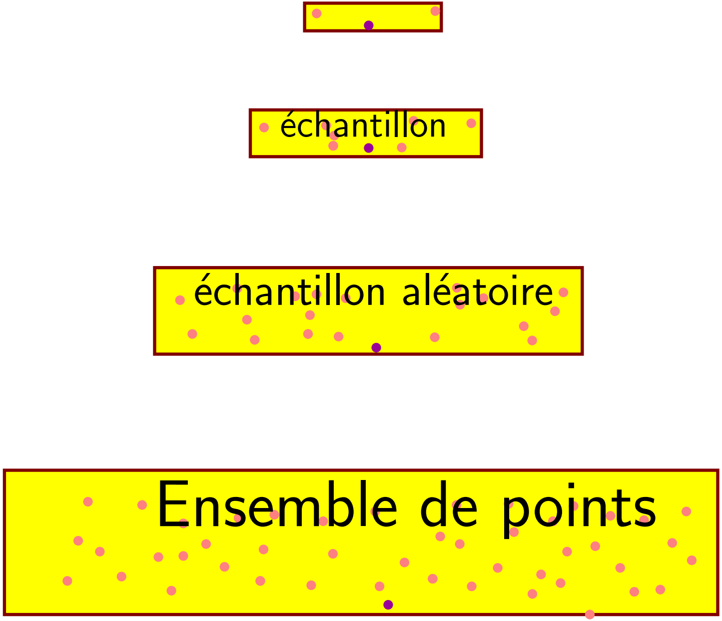
Hiérarchie de Delaunay

pointeurs

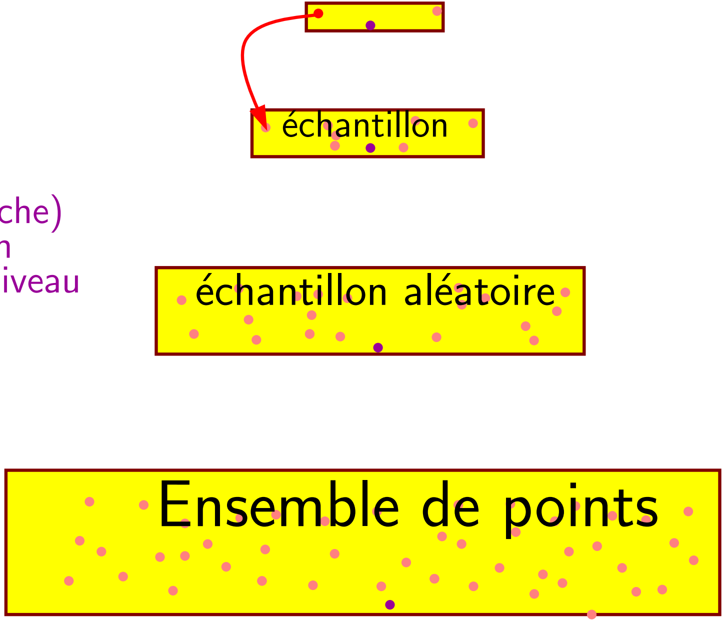


Hiérarchie de Delaunay

requête



Hiérarchie de Delaunay



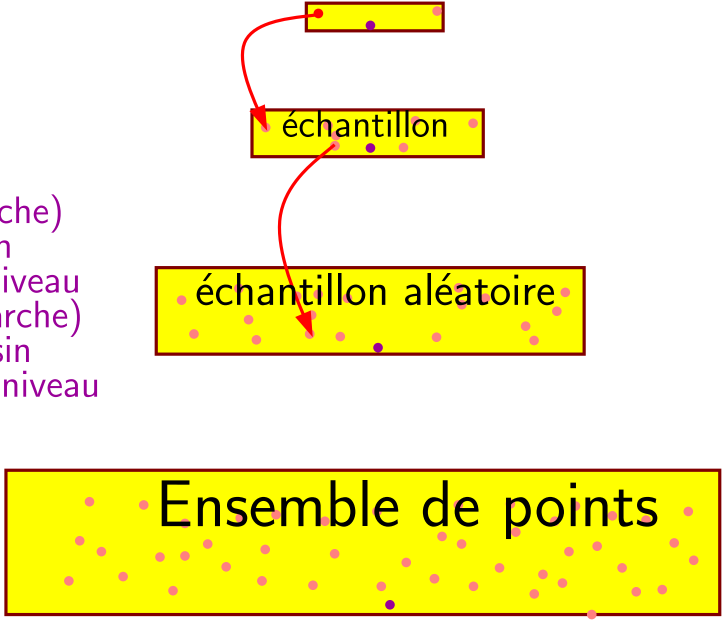
requête

localisation (marche)
plus proche voisin
changement de niveau

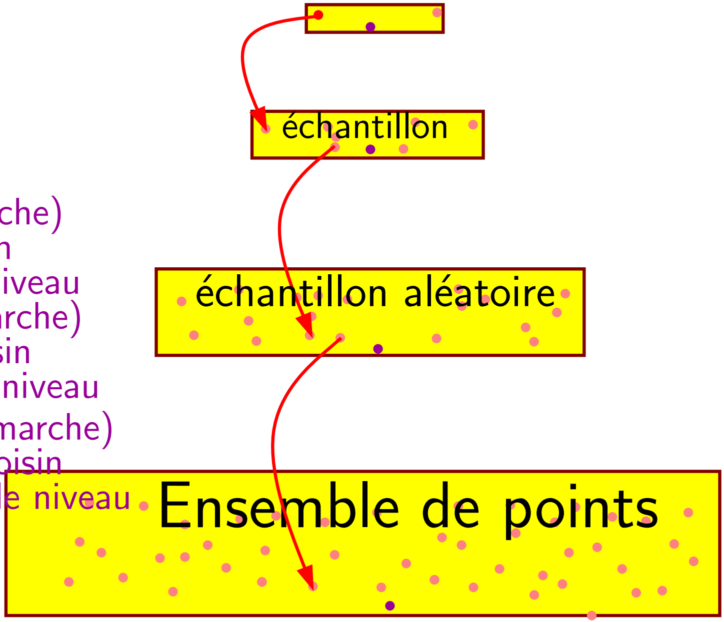
Hiérarchie de Delaunay

requête

- localisation (marche)
- plus proche voisin
- changement de niveau
  - localisation (marche)
  - plus proche voisin
  - changement de niveau



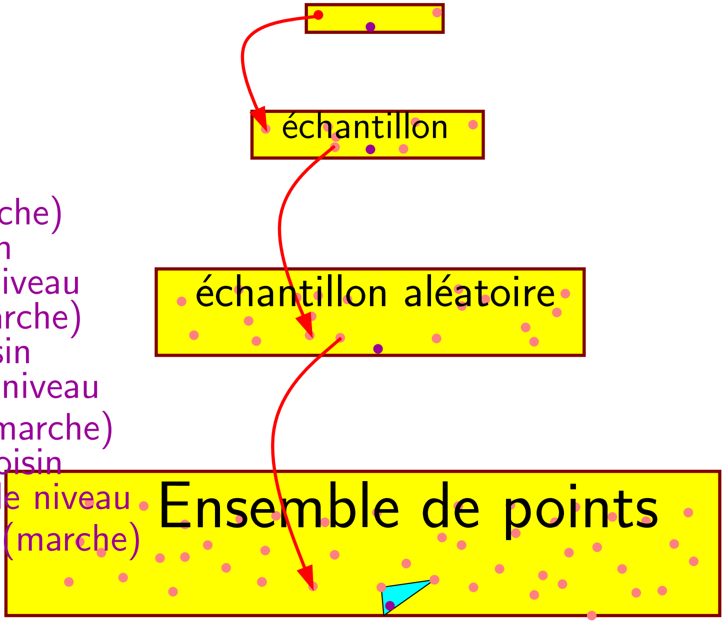
Hiérarchie de Delaunay



requête

- localisation (marche)
- plus proche voisin
- changement de niveau
  - localisation (marche)
  - plus proche voisin
  - changement de niveau
    - localisation (marche)
    - plus proche voisin
    - changement de niveau

Hiérarchie de Delaunay



requête

localisation (marche)
plus proche voisin
changement de niveau
localisation (marche)
plus proche voisin
changement de niveau
localisation (marche)
plus proche voisin
changement de niveau
localisation (marche)

Points uniformément distribués

échantillonage :

Points uniformément distribués

échantillonage : $\alpha = \dfrac{n}{k}$

Points uniformément distribués

échantillonage : $\alpha = \frac{n}{k}$

$$\sqrt{\frac{n}{k}} + \sqrt{\frac{k}{k'}} + \dots$$

Points uniformément distribués

échantillonage : $\alpha = \dfrac{n}{k}$

$$\sqrt{\frac{n}{k}} + \sqrt{\frac{k}{k'}} + \ldots = \sqrt{\alpha} + \sqrt{\alpha} + \ldots = \log_{\alpha} n \cdot \sqrt{\alpha}$$

Points uniformément distribués

échantillonage : $\alpha = \dfrac{n}{k}$

$$\sqrt{\frac{n}{k}} + \sqrt{\frac{k}{k'}} + \ldots = \sqrt{\alpha} + \sqrt{\alpha} + \ldots = \log_\alpha n \cdot \sqrt{\alpha}$$

Si points NON uniformément distribués

Points uniformément distribués

échantillonage : $\alpha = \dfrac{n}{k}$

$$\sqrt{\frac{n}{k}} + \sqrt{\frac{k}{k'}} + \ldots = \sqrt{\alpha} + \sqrt{\alpha} + \ldots = \log_\alpha n \cdot \sqrt{\alpha}$$

Si points NON uniformément distribués

$$\frac{n}{k} + \frac{k}{k'} + \ldots \qquad = \log_\alpha n \cdot \alpha$$

Points uniformément distribués

échantillonage : $\alpha = \dfrac{n}{k}$

$$\sqrt{\frac{n}{k}} + \sqrt{\frac{k}{k'}} + \ldots = \sqrt{\alpha} + \sqrt{\alpha} + \ldots = \log_\alpha n \cdot \sqrt{\alpha}$$

Si points NON uniformément distribués

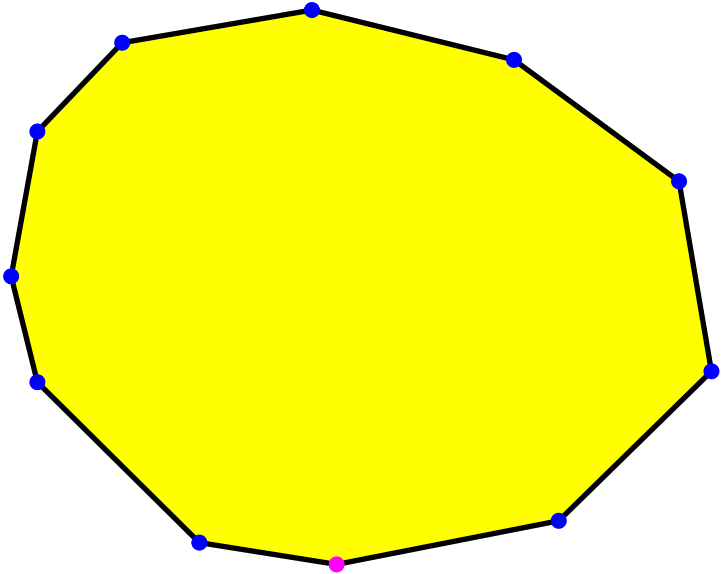
$$\frac{n}{k} + \frac{k}{k'} + \ldots \qquad = \log_\alpha n \cdot \alpha$$

$\boxed{\text{Delaunay en } O(n \log n) \ (\forall \text{ distribution})}$

Algorithmes randomisés accélérés

la triangulation de Delaunay

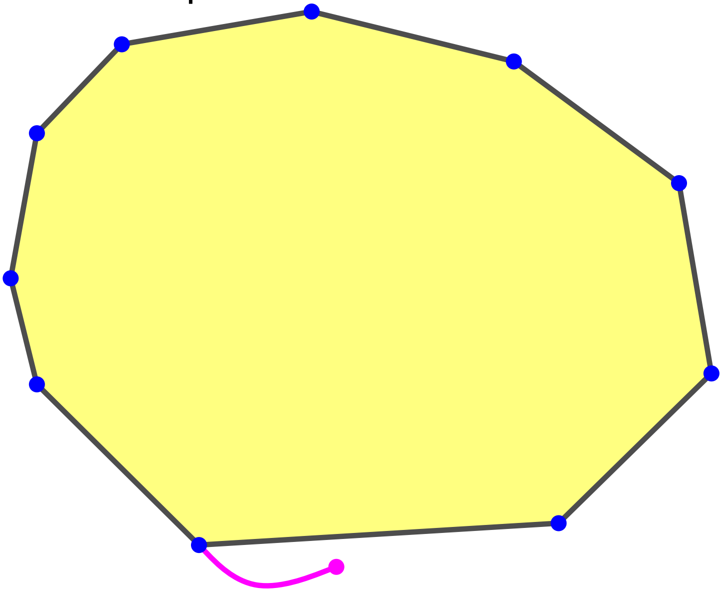
d'un polygone convexe



Selection d'un point aléatoire

Suppression du convexe

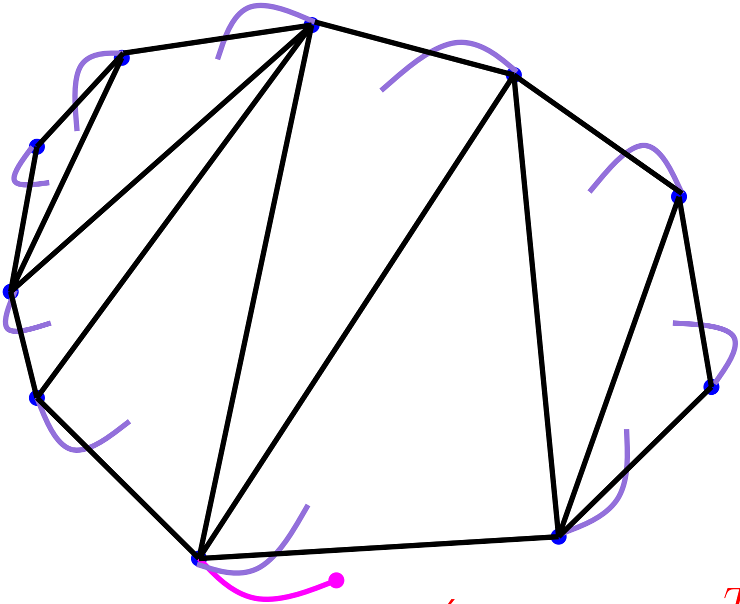
Lien vers le sommet précédent



$O(1)$

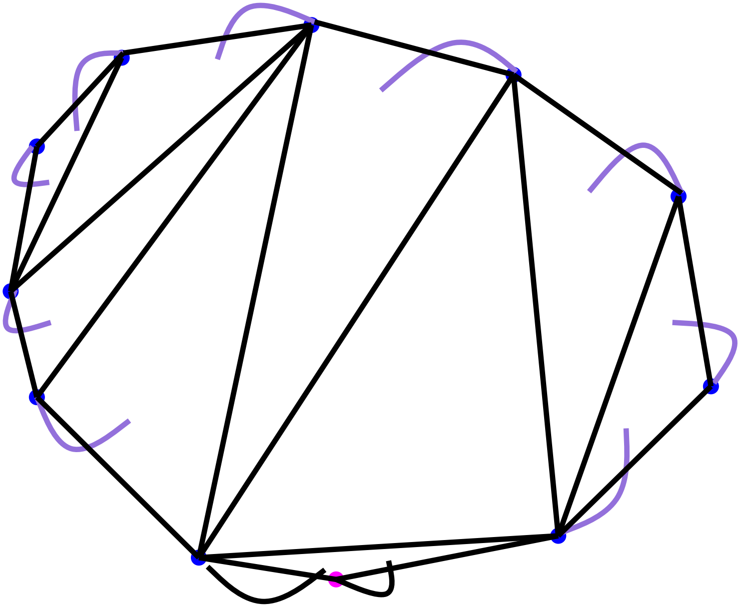
Calcul récursif de Delaunay

Avec les pointeurs



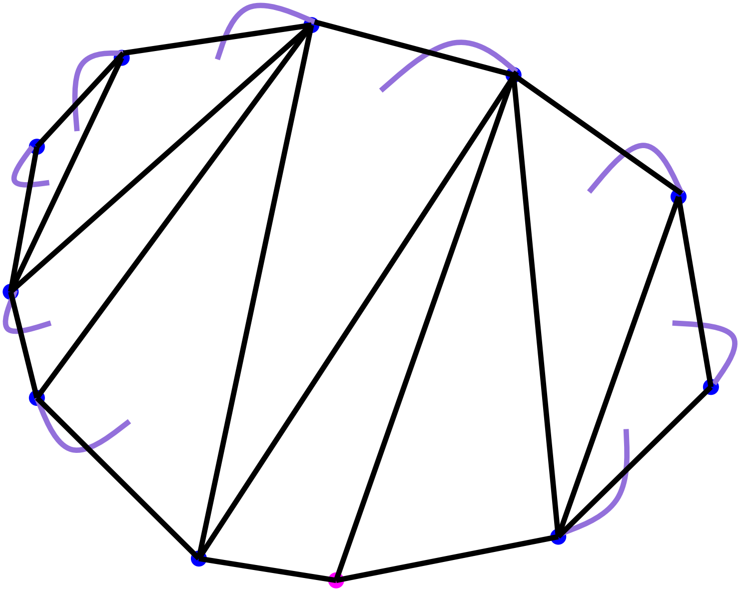
$T(n-1)$

Insertion du point supprimé



$O(1)$

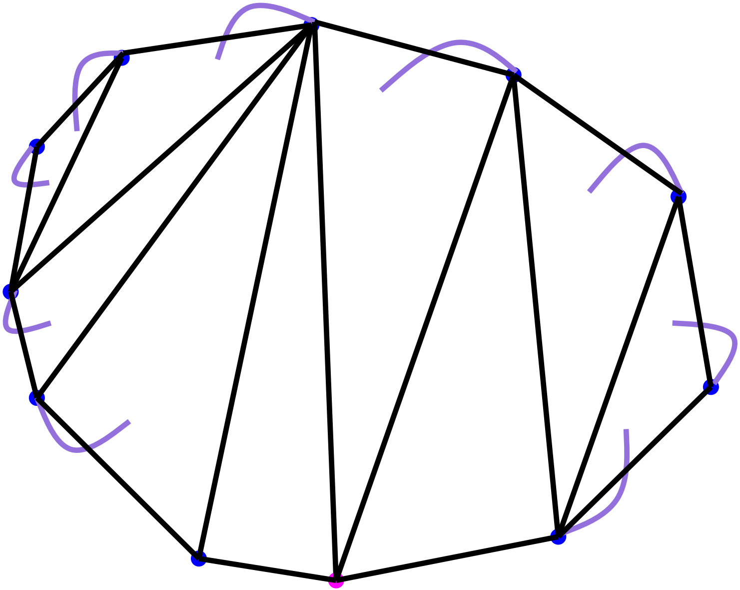
Bascule de Delaunay



Anlyse randomisée

$O(1)$

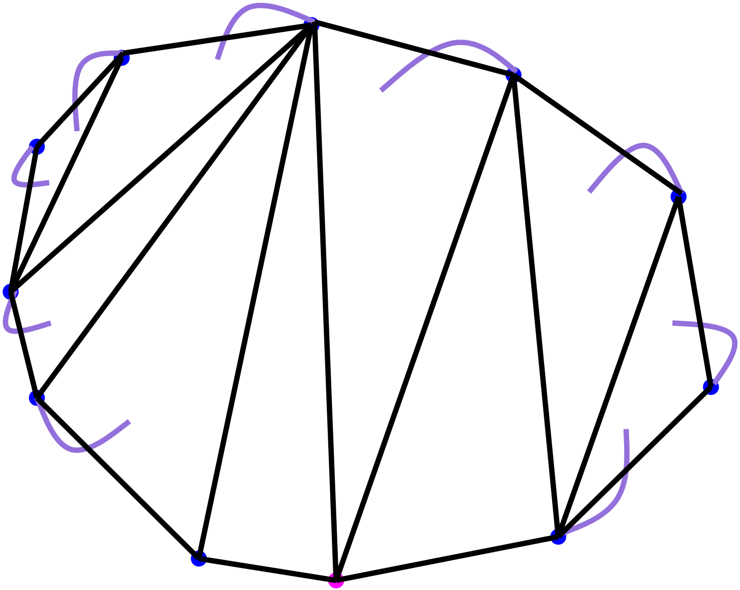
Bascule de Delaunay



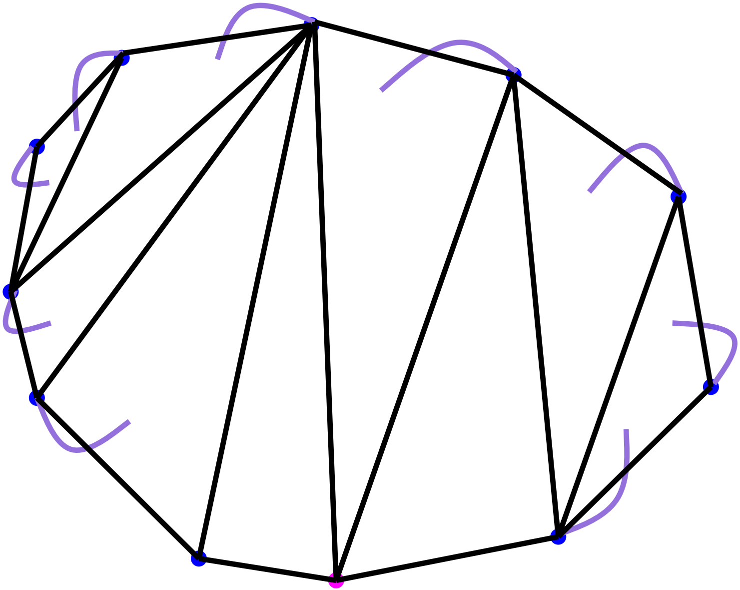
Anlyse randomisée

$O(1)$

$$T(n) = T(n-1) + O(1)$$



$T(n) = O(n)$



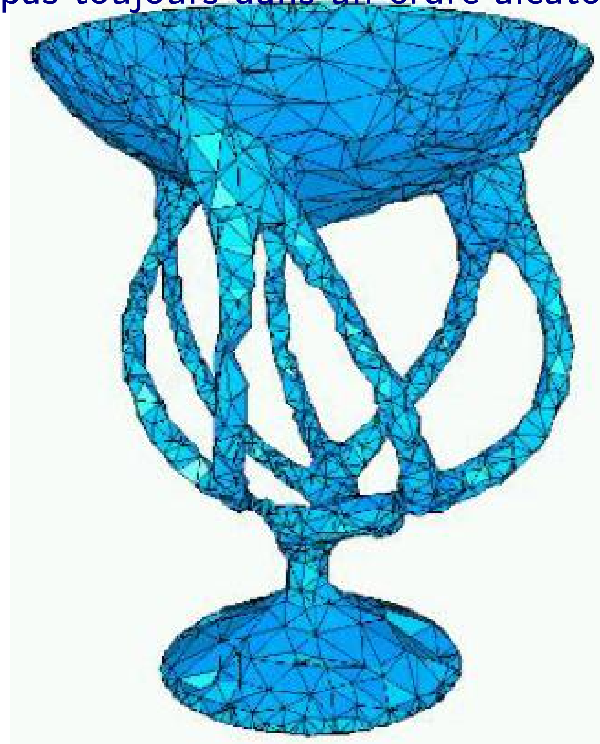
Algorithmes randomisés

la triangulation de Delaunay

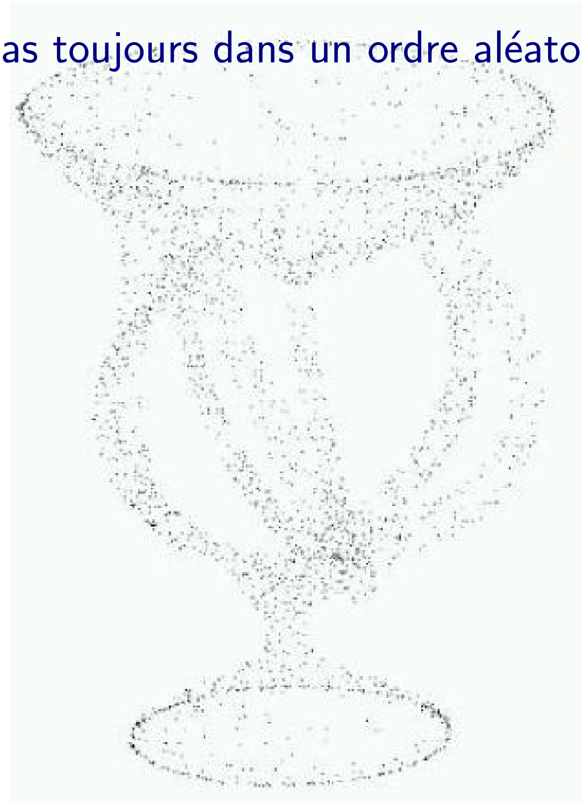
le tampon mélangeur

Les points sont pas toujours dans un ordre aléatoire

Les points sont pas toujours dans un ordre aléatoire



Les points sont pas toujours dans un ordre aléatoire



Les points sont pas toujours dans un ordre aléatoire

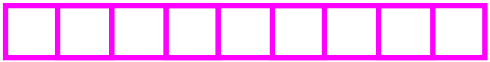
Bien secouer avant d'ingurgiter !

Les points sont pas toujours dans un ordre aléatoire

Bien secouer avant d'ingurgiter !

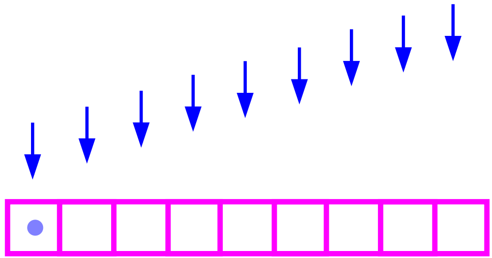
Il faut déjà avoir tous les points

Alternative : le tampon mélangeur



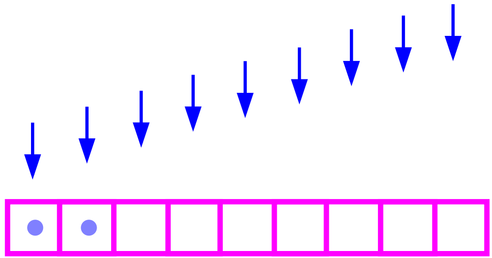
Alternative : le tampon mélangeur

On remplit le tampon avec k données



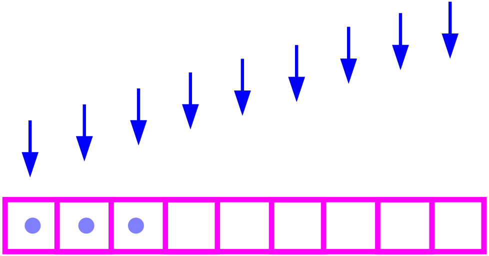
Alternative : le tampon mélangeur

On remplit le tampon avec k données



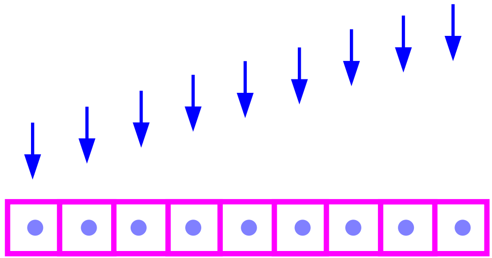
Alternative : le tampon mélangeur

On remplit le tampon avec k données



Alternative : le tampon mélangeur

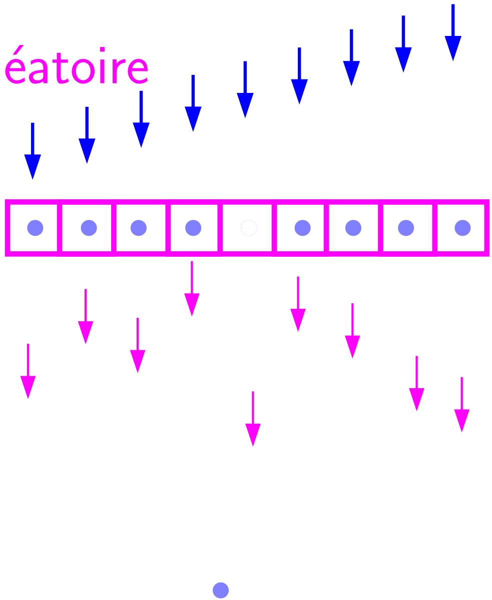
On remplit le tampon avec k données



Alternative : le tampon mélangeur

On remplit le tampon avec k données

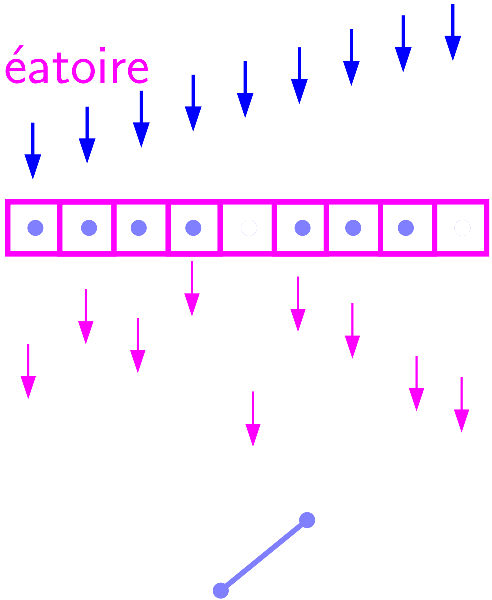
On vide dans un ordre aléatoire



Alternative : le tampon mélangeur

On remplit le tampon avec k données

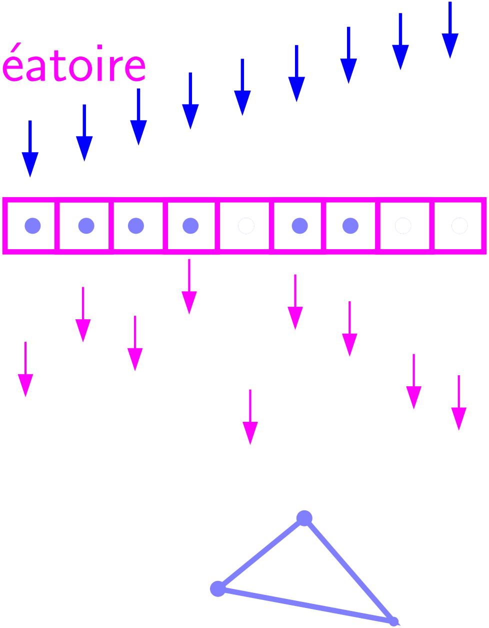
On vide dans un ordre aléatoire



Alternative : le tampon mélangeur

On remplit le tampon avec k données

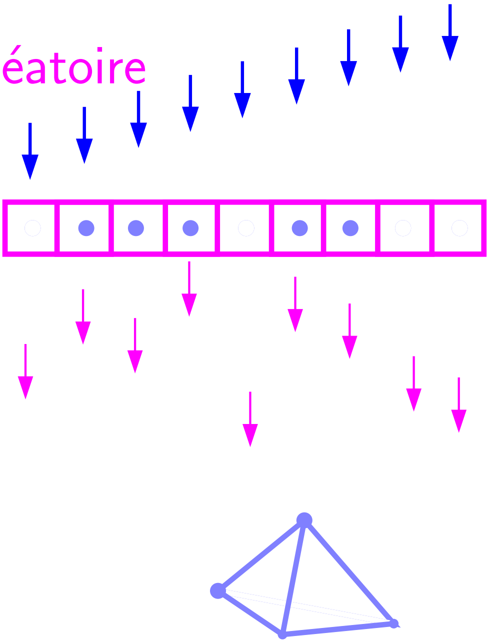
On vide dans un ordre aléatoire



Alternative : le tampon mélangeur

On remplit le tampon avec k données

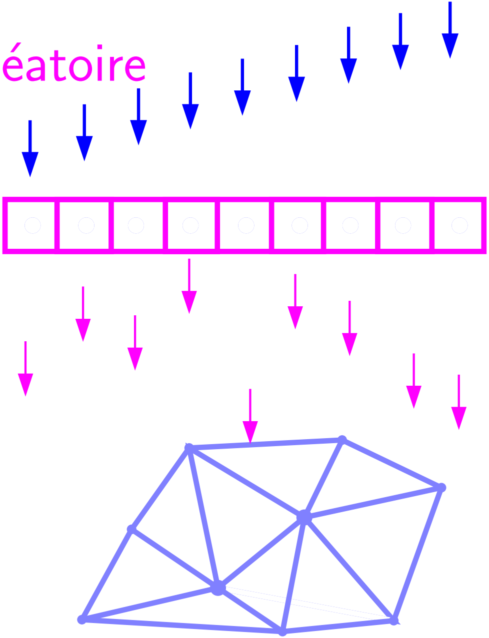
On vide dans un ordre aléatoire



Alternative : le tampon mélangeur

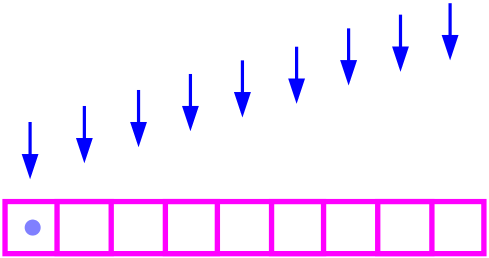
On remplit le tampon avec k données

On vide dans un ordre aléatoire

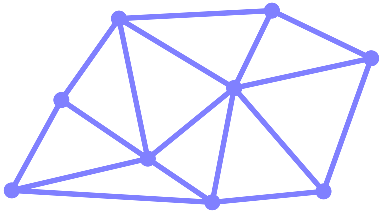


Alternative : le tampon mélangeur

On remplit le tampon avec k données

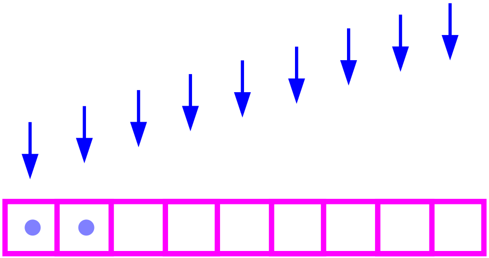


On recommence

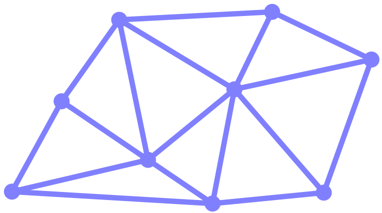


Alternative : le tampon mélangeur

On remplit le tampon avec k données

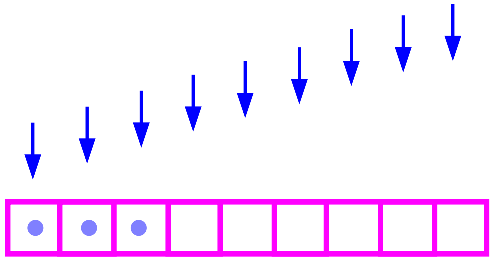


On recommence

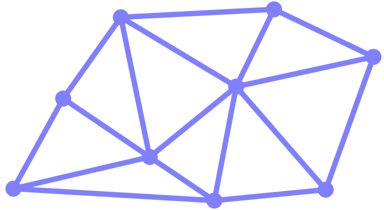


Alternative : le tampon mélangeur

On remplit le tampon avec k données

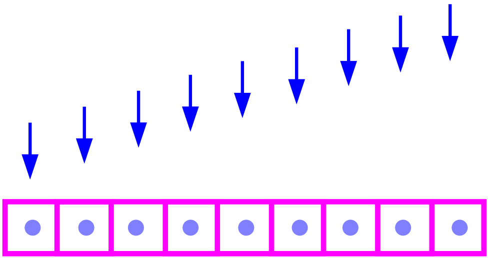


On recommence

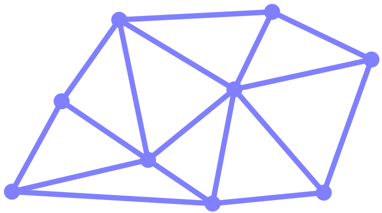


Alternative : le tampon mélangeur

On remplit le tampon avec k données



On recommence

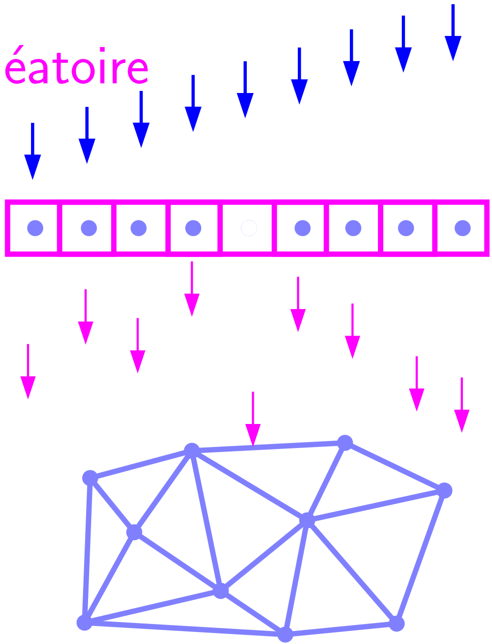


Alternative : le tampon mélangeur

On remplit le tampon avec k données

On vide dans un ordre aléatoire

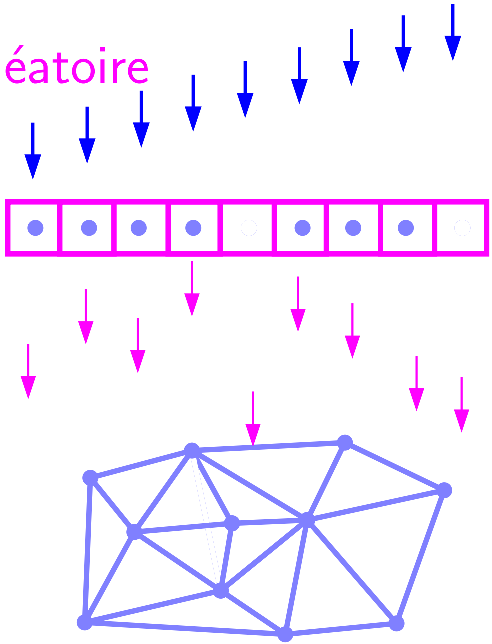
On recommence



Alternative : le tampon mélangeur

On remplit le tampon avec k données

On vide dans un ordre aléatoire

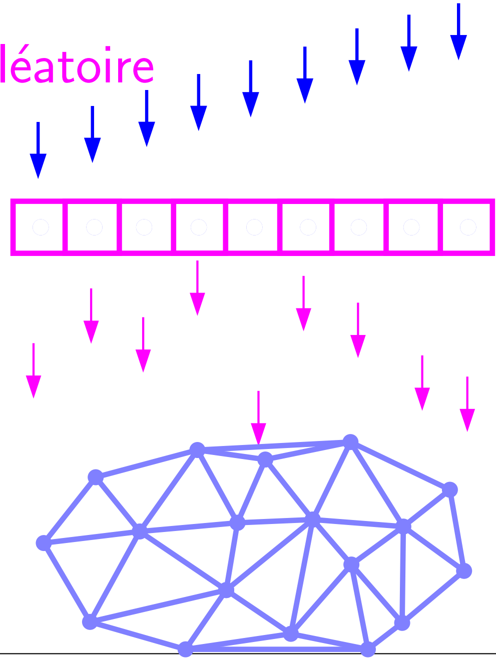


On recommence

Alternative : le tampon mélangeur

On remplit le tampon avec k données

On vide dans un ordre aléatoire



On recommence

T le dernier tampon (taille k)

coût du dernier point (degré)

$$= \frac{1}{k} \sum_{s \in T} \text{degré(s)}$$

T le dernier tampon (taille k)

coût du dernier point (degré)

$$= \frac{1}{k} \sum_{s \in T} \text{degré}(\text{s})$$

$$\sum_{s \in T} \text{degré}(\text{s}) \leq \sum_{s \in S} \text{degré}(\text{s}) < 6n$$

T le dernier tampon (taille k)

coût du dernier point (degré)

$$= \frac{1}{k} \sum_{s \in T} \text{degré(s)}$$

$$\sum_{s \in T} \text{degré(s)} \leq \sum_{s \in S} \text{degré(s)} < 6n$$

$$\leq \frac{6n}{k}$$

coût du dernier point (degré) $\leq \dfrac{6n}{k}$

coût du dernier point (degré) $\leq \frac{6n}{k}$

Coût de T, le $p+1$ème tampon

$$\leq \sum_{i=1}^{k} \frac{6(pk+i)}{i} = O(pk \log k)$$

coût du dernier point (degré) $\leq \dfrac{6n}{k}$

Coût de T, le $p+1^{\text{ème}}$ tampon

$$\leq \sum_{i=1}^{k} \frac{6(pk+i)}{i} = O(pk\log k)$$

Coût total

$$\leq O\left(\sum_{p=0}^{\lfloor \frac{n}{k} \rfloor} pk\log k\right)$$

$$\leq O(k\log k)\sum_{p=0}^{\lfloor \frac{n}{k} \rfloor} p \leq O\left(k\log k\left(\frac{n}{k}\right)^2\right)$$

coût du dernier point (degré) $\leq \dfrac{6n}{k}$

Coût de T, le $p+1$ème tampon

$$\leq \sum_{i=1}^{k} \frac{6(pk+i)}{i} = O(pk \log k)$$

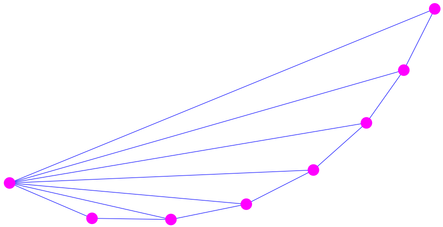
Coût total

$$\leq O\left(\sum_{p=0}^{\lfloor \frac{n}{k} \rfloor} pk \log k\right)$$

$$\leq O(k \log k) \sum_{p=0}^{\lfloor \frac{n}{k} \rfloor} p \leq \boxed{\Theta\left(\frac{n^2 \log k}{k}\right)}$$

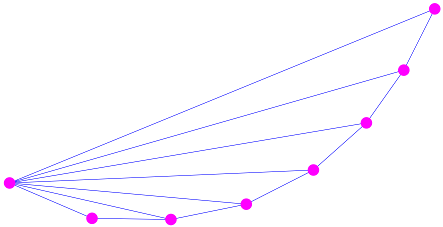
Cas le pire

$k = 1$

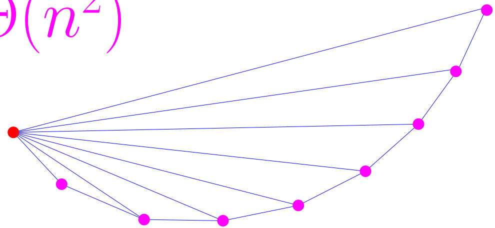


Cas le pire

$k = 1$



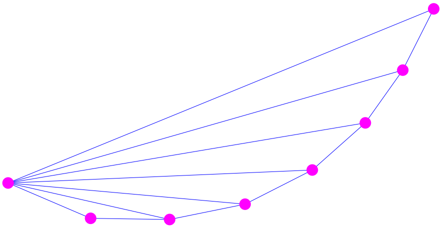
$\Theta(n^2)$

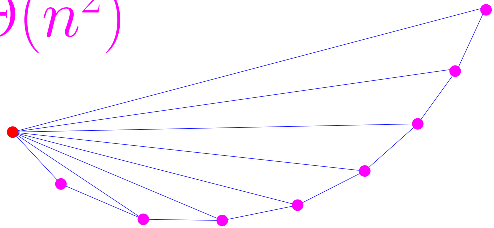


Cas le pire

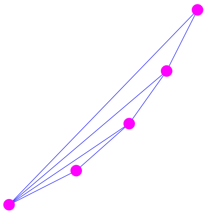
$k = 1$

$\Theta(n^2)$





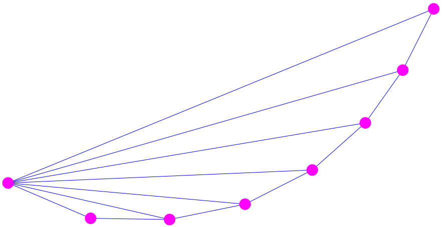
$k = \frac{n}{2}$

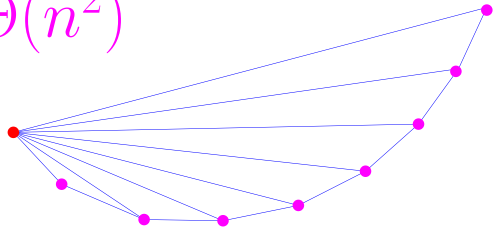


Cas le pire

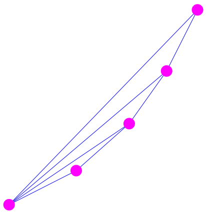
$$k = 1$$

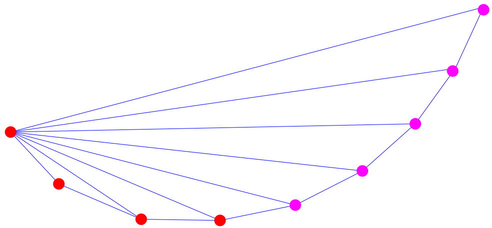
$$\Theta(n^2)$$





$$k = \frac{n}{2} \quad \Theta(n \log n)$$





C'était le coût d'insertion

C'était le coût d'insertion

Coût de localisation (depends algo)

C'était le coût d'insertion

Coût de localisation (depends algo)

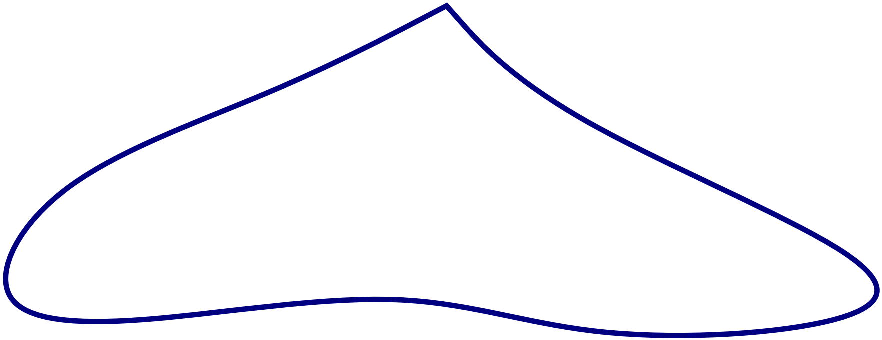
	[Delaunay tree]	Delaunay Hierarchy
Cas le pire	$O(n^3)$	$O(n^2 \log n)$
Randomisé	$O(n \log n)$	$O(n \log n)$
tampon mélangeur	$O(n^{3\frac{\log k}{k}})$	$O(n^{2\frac{\log k}{k}})$

Algorithmes randomisés accélérés

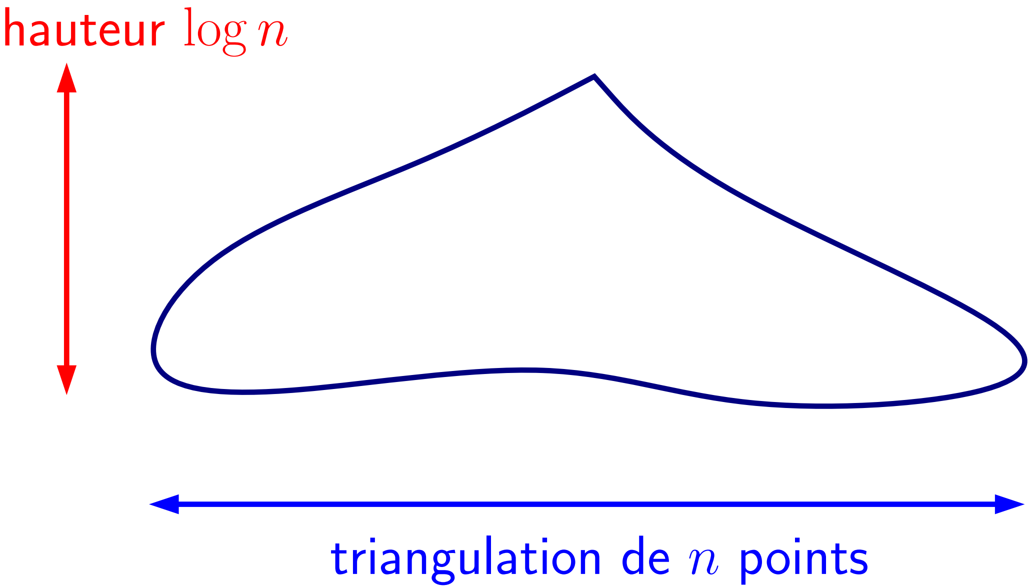
la triangulation de Delaunay

en connaissant l'arbre couvrant minimal

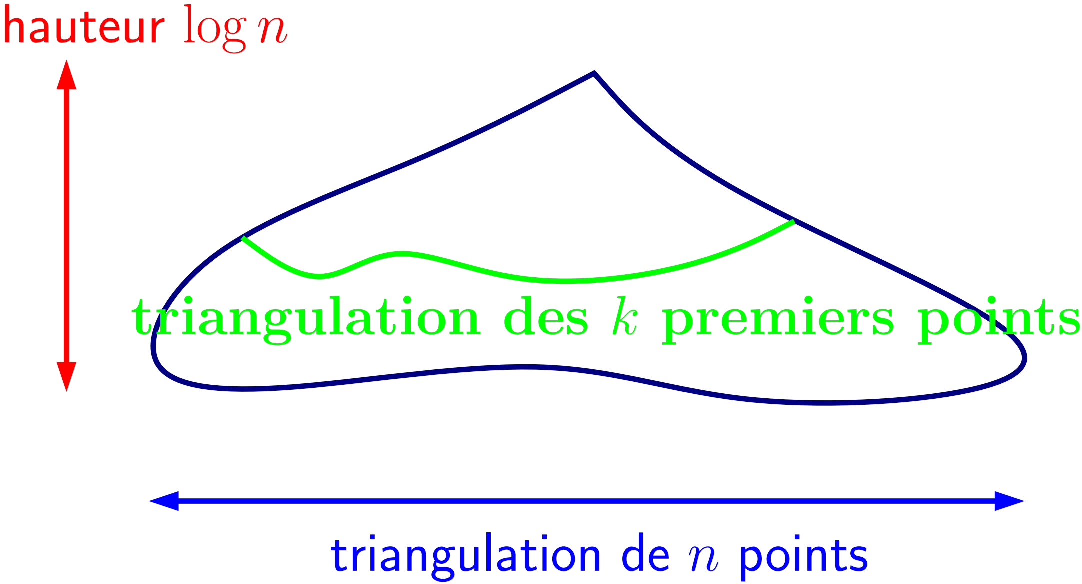
Arbre de Delaunay



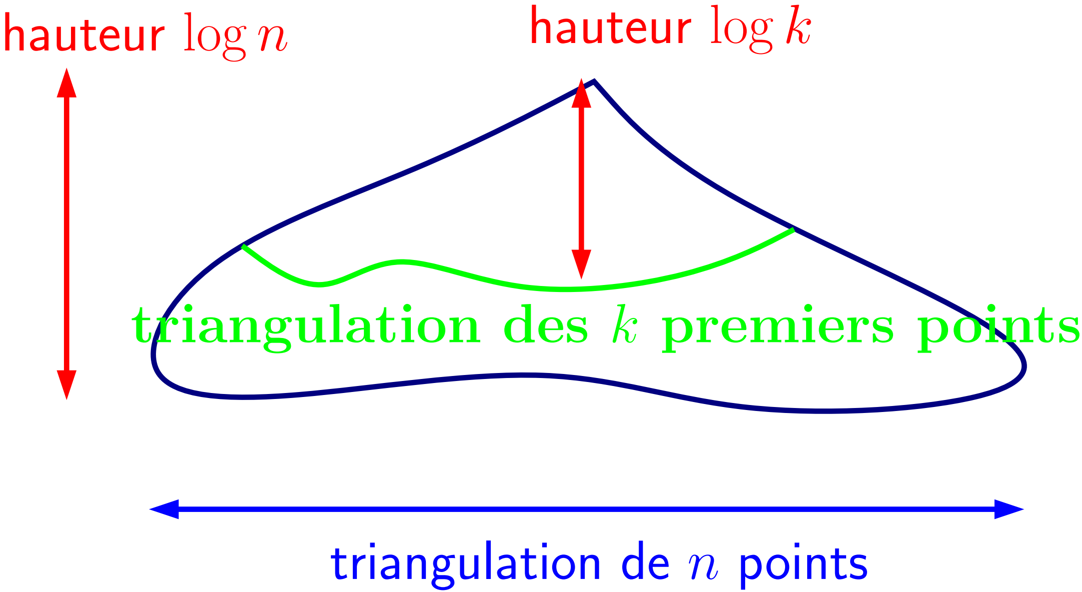
Arbre de Delaunay



Arbre de Delaunay

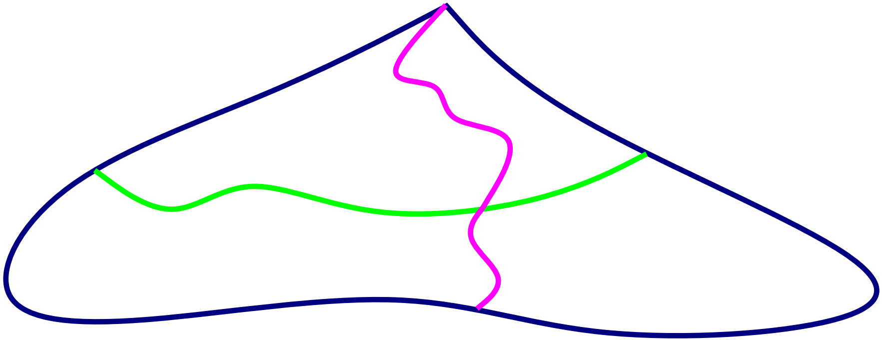


Arbre de Delaunay

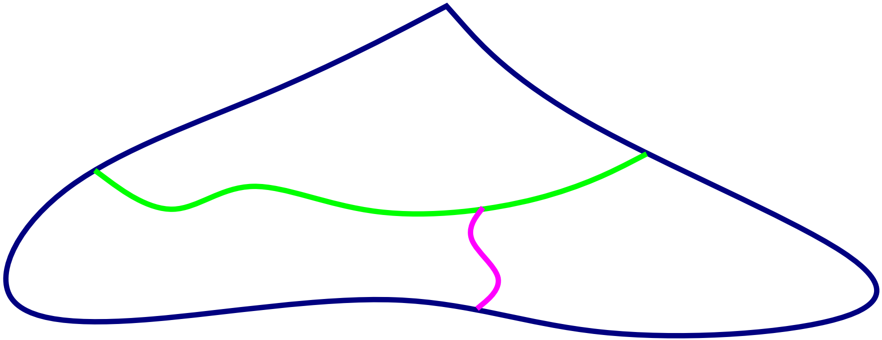


Arbre de Delaunay

localisation



Arbre de Delaunay



$$\log n - \log k = \log \frac{n}{k}$$

Construction avec les k premiers points

Construction avec les k premiers points

Pour chaque point non inséré:

localisation conjointe

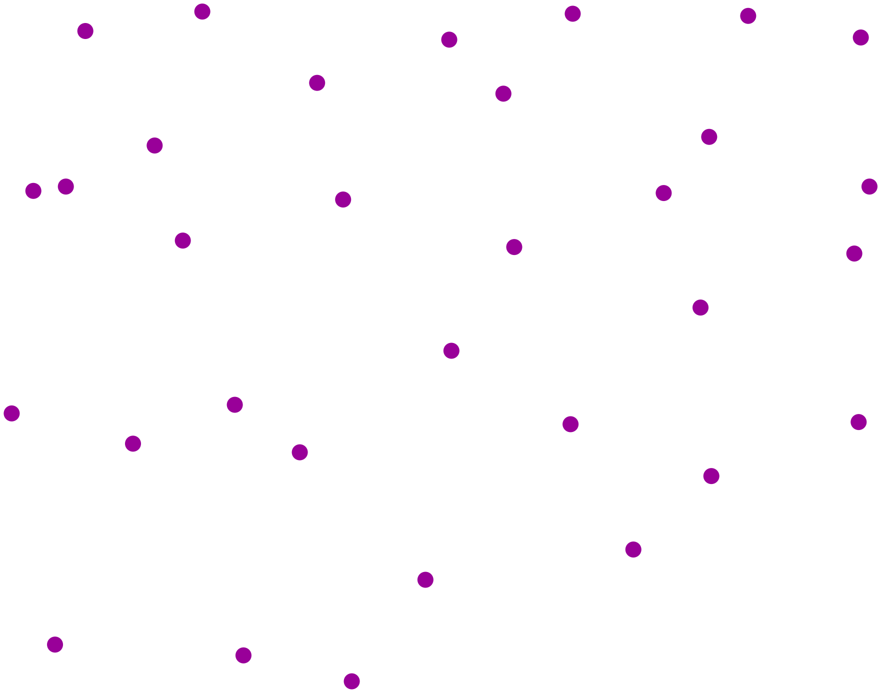
Construction avec les k premiers points

Pour chaque point non inséré:

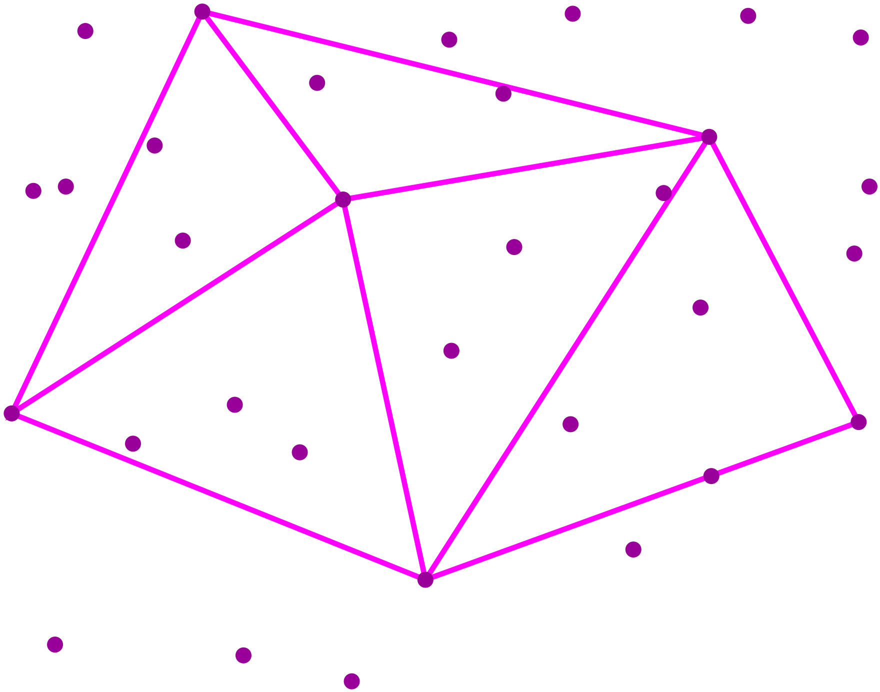
localisation conjointe

Localisation partielle pour continuer

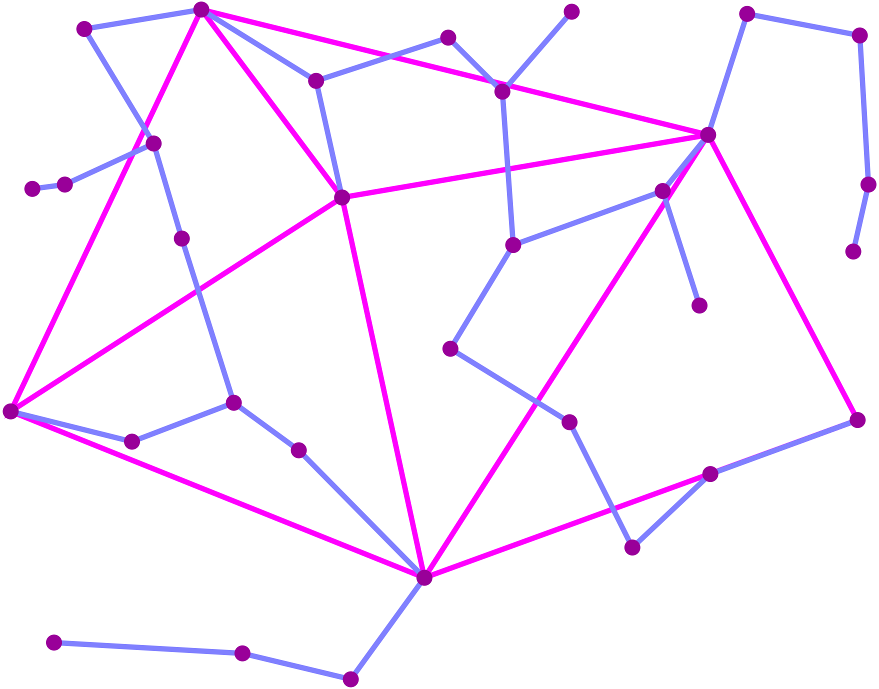
localisation conjointe



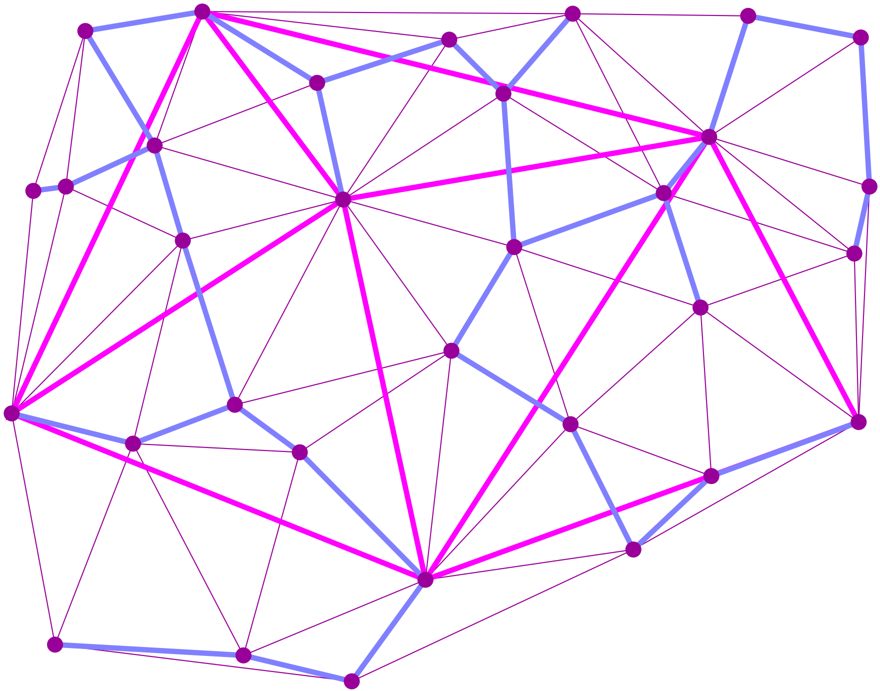
localisation conjointe



localisation conjointe



localisation conjointe



$\frac{n}{\log n}$ premiers points

$\frac{n}{\log n}$ premiers points

$\log n$ par point

$\dfrac{n}{\log n}$ premiers points

$\log n$ par point

$\longrightarrow$ $O(n)$

$\dfrac{n}{\log n}$ premiers points

$\log n$ par point

$\longrightarrow \quad O(n)$

localisation conjointe

$\dfrac{n}{\log n}$ premiers points

$\log n$ par point

$\longrightarrow \quad O(n)$

localisation conjointe

$\longrightarrow \quad O(n)$

$\frac{n}{\log n}$ premiers points $\longrightarrow$ $O(n)$

$\frac{n}{\log \log n}$ premiers points

$\dfrac{n}{\log n}$ premiers points $\longrightarrow$ $O(n)$

$\dfrac{n}{\log \log n}$ premiers points

$\log \log n$ par point

$\dfrac{n}{\log n}$ premiers points $\longrightarrow$ $O(n)$

$\dfrac{n}{\log \log n}$ premiers points

$\log \log n$ par point

$\longrightarrow$ $O(n)$

$\dfrac{n}{\log n}$ premiers points $\longrightarrow$ $O(n)$

$\dfrac{n}{\log\log n}$ premiers points

$\log\log n$ par point

$\longrightarrow$ $O(n)$

localisation conjointe

$\dfrac{n}{\log n}$ premiers points $\longrightarrow$ $O(n)$

$\dfrac{n}{\log \log n}$ premiers points

$\log \log n$ par point

$\longrightarrow$ $O(n)$

localisation conjointe

$\longrightarrow$ $O(n)$

$\dfrac{n}{\log n}$ premiers points $\longrightarrow O(n)$

$\longrightarrow O(n)$

$\dfrac{n}{\log \log \log n}$ premiers points

$\dfrac{n}{\log n}$ premiers points $\longrightarrow O(n)$

$\longrightarrow O(n)$

$\dfrac{n}{\log \log \log n}$ premiers points

$\dfrac{n}{\log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log \log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log \log \log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log \log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log \log \log n}$ premiers points $\longrightarrow O(n)$

$\cdots\cdot$

$\dfrac{n}{\log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log\log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log\log\log n}$ premiers points $\longrightarrow O(n)$

$\cdots\,.$

jusqu'à $\log\log\ldots\log n \leq 1$

$\dfrac{n}{\log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log \log n}$ premiers points $\longrightarrow O(n)$

$\dfrac{n}{\log \log \log n}$ premiers points $\longrightarrow O(n)$

$\bullet \; \bullet \; \bullet \; \bullet$

jusqu'à $\log \log \ldots \log n \leq 1$

$O(n \log^{\star} n)$

$$O(n \log^\star n)$$

$$\log^\star 2 = 1$$

$$\log^\star 4 = 2$$

$$\log^\star 16 = 3$$

$$\log^\star 65536 = \log^\star 2^{16} = 4$$

$$\log^\star \textsf{très grand} = \log^\star 2^{65536} = 5$$

Algorithmes randomisés

la triangulation de Delaunay

lazy cleaning

C’est tout pour aujourd’hui

C’est tout pour aujourd’hui