Conception et Pratique de l'Algorithmique

http://www-apr.lip6.fr/~buixuan/cpa2016

Binh-Minh Bui-Xuan

Paris, Janvier 2016

Cours 2 : enveloppe convexe

Rappel cours + tme 1 :

- collision : cas le plus simple est la collision des cercles
- problème CercleMin : nuage de points → cercle
- algorithme naïf : complexité $O(n^4)$
- techniques : incrémental (Ritter), précalcul (Akl-Toussaint)

Cours 2 : enveloppe convexe

RAPPEL COURS + TME 1 :

- collision : cas le plus simple est la collision des cercles
- problème CERCLEMIN : nuage de points → cercle
- algorithme naïf : complexité $O(n^4)$
- techniques : incrémental (Ritter), précalcul (Akl-Toussaint)

AUJOURD'HUI :

- collision : cas de polygones convexes (esthétique !)
- problème ENVCONVEXE : nuage de points → polygone convexe
- algorithme naïf : complexité $O(n^3)$
- techniques : précalcul (pixel, Akl-Toussaint), décomposition
- algorithmes : Jarvis, Graham (+variants), Chan, QuickHull

Sources wiki (à lire avec recul)

THE GAME PROGRAMMING WIKI :

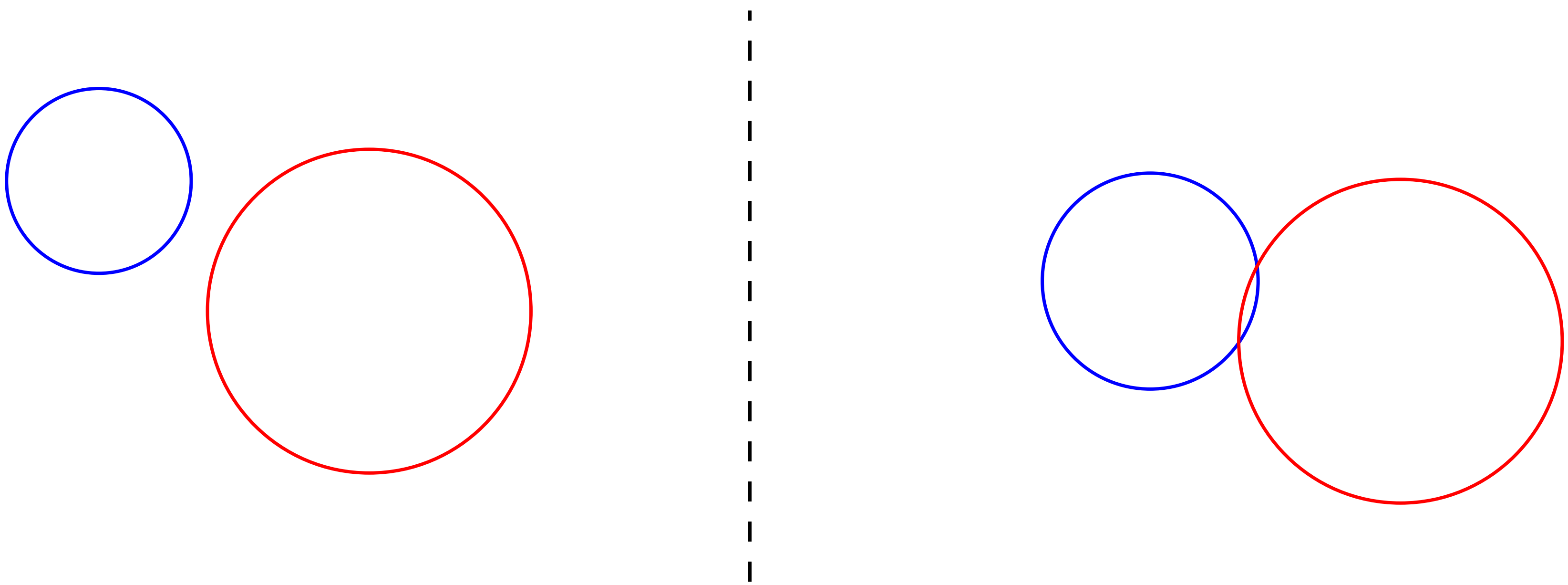
– http://content.gpwiki.org/index.php/Polygon_Collision

WIKIPEDIA :

– http://en.wikipedia.org/wiki/Convex_hull_algorithms

Collision d'objets géométriques

QUESTION : touché ?

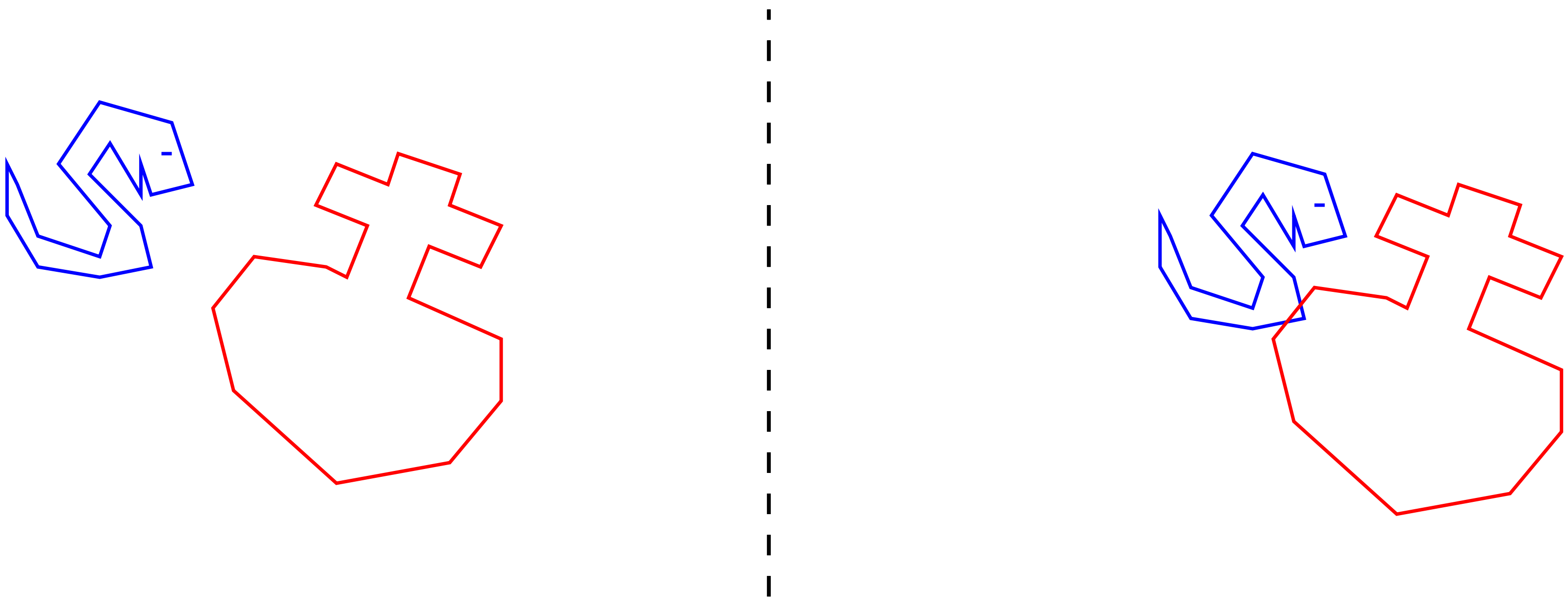


Collision d'objets géométriques

QUESTION : touché ?

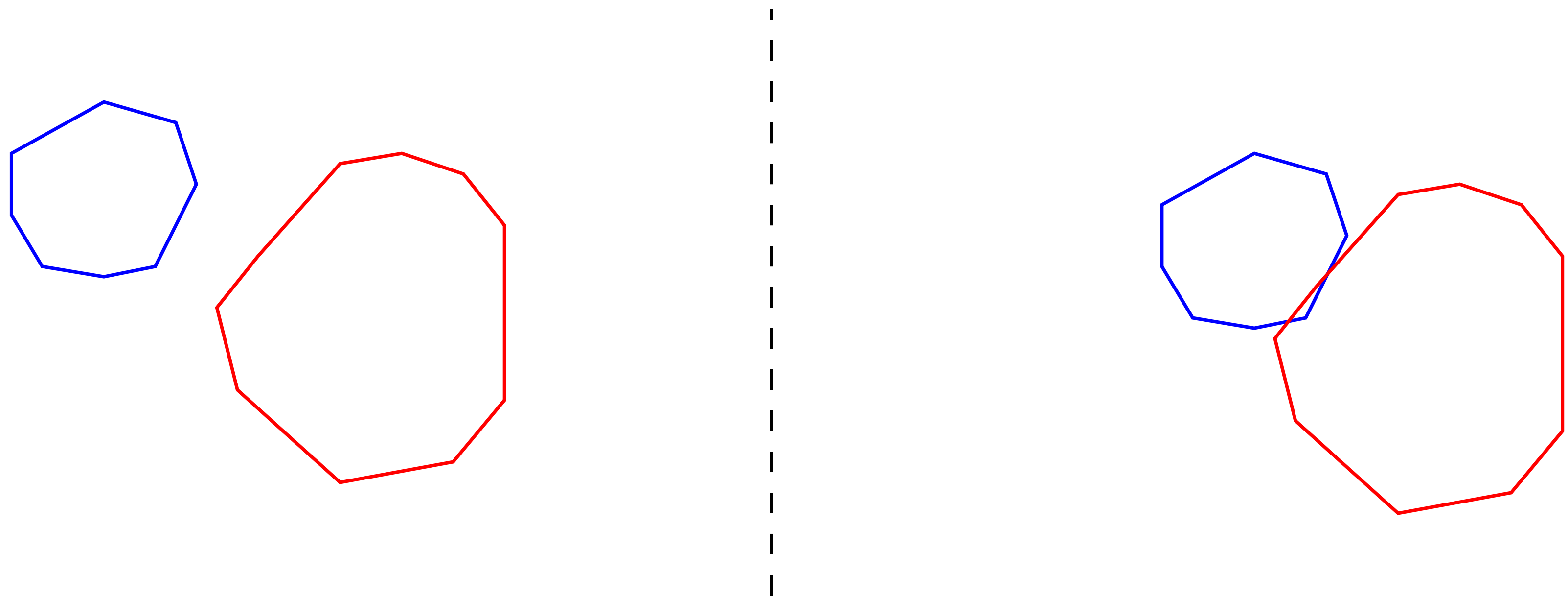
Collision d'objets géométriques

QUESTION : touché ?



Collision d'objets géométriques

Question : touché ?



Polygone convexe

DÉFINITIONS :

– *polygone convexe* : polygone tel que tout segment joinant deux points du polygone appartient au polygone

– *côté* d'un polygone convexe : segment maximal du polygone tel que tout autre point du polygone appartient au même demi-plan défini par le segment

– *coin* d'un polygone convexe : extrémité d'un côté

– *contour positif* d'un polygone convexe : l'ordre cyclique de ses coins selon le sens trigonométrique

Exercice : collision entre convexes

EXERCICE : Soient deux polygones convexes `p1` et `p2`. Montrer qu'il n'y a pas de collision entre `p1` et `p2` si et seulement si une des deux conditions suivantes est vérifiée :

– il existe un côté de `p1` tel que tout coin de `p2` est "de l'autre côté" des coins de `p1` dans les demi-plans définis par ce côté

– il existe un côté de `p2` tel que tout coin de `p1` est "de l'autre côté" des coins de `p2` dans les demi-plans définis par ce côté

QUESTION : implantation ?

Notes pour révision : algorithme

```
pour tout côté pq de p1
   r ← un coin de p1 distinct de p et de q
   côtéSéparant ← true
   pour tout coin s de p2
      si s est du même côté que r par rapport à (pq) alors
         côtéSéparant ← false
   si côtéSéparant = true alors retourner pas_de_collision
pour tout côté pq de p2
   r ← un coin de p2 distinct de p et de q
   côtéSéparant ← true
   pour tout coin s de p1
      si s est du même côté que r par rapport à (pq) alors
         côtéSéparant ← false
   si côtéSéparant = true alors retourner pas_de_collision
retourner collision
```

Problème de l'enveloppe convexe

IN : `Points`, une liste de coordonnées de points en 2D

OUT : `Enveloppe`, une liste d'éléments de `Points` représentant le contour positif de l'enveloppe convexe de `Points`

DÉFINITION : l'enveloppe convexe d'un ensemble de points est le plus petit polygone convexe contenant ces points

Solution naïve

PRINCIPE : pour toute paire de points dans `Points`, vérifier s'elle forme un côté de l'enveloppe convexe de `Points` en testant si tout autre point de `Points` sont "du même côté" des demi-plans définis par le segment passant par la paire de points

QUESTION : complexité ? implantation ?

Notes pour révision : voir TME2.

Exercice : estimation du temps

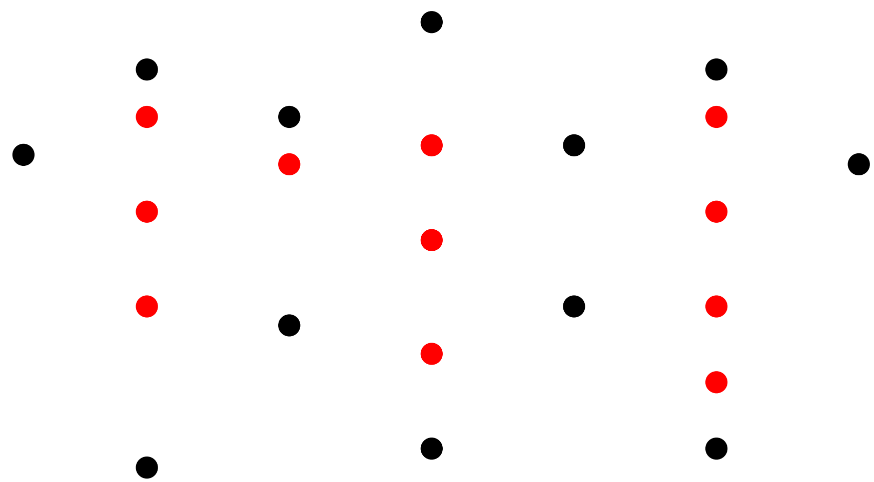
QUESTION : un ordinateur de l'ordre du Giga-Hertz exécutant un algorithme en $O(n^3)$, avec $n = 10000$, quel est le temps d'exécution attendu (en ordre de grandeur) ?

- algorithme en $O(n \log n)$?
- algorithme en $O(n \log h)$, avec $h << n$?
- algorithme en $O(n + n' \log h)$, avec $n' << n$ et $h << n$?

Enveloppe convexe : précalculs

Elimination par pixel

IDÉE : si trois points de `Points` ont le même abscisse, alors au plus deux de ces points appartiennent à l'enveloppe convexe de `Points`

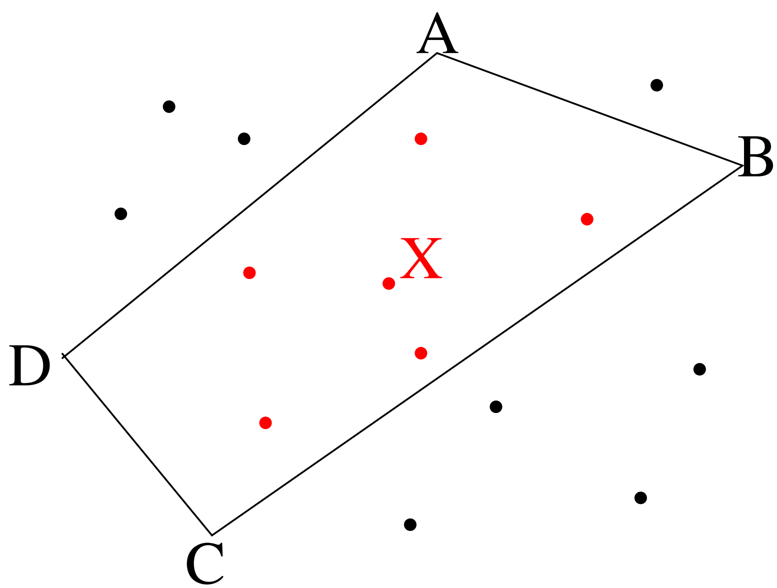


Notes pour révision : algorithme

1. `ymin` $\leftarrow$ tableau de points de taille assez grande
2. pour tout `p` dans `Points`
3.   si `ymin[p.x]` $=$ `NULL` ou si `ymin[p.x].y` $< p.y$
4.   `ymin[p.x]` $\leftarrow$ `p`
5. similairement avec `ymax`
6. parcourir `ymin` et `ymax` et retourner les éléments non `NULL`

Filtrage Akl-Toussaint

IDÉE : soient `A`, `B`, `C`, `D` les points les plus au Nord, Est, Sud, Ouest parmi `Points`, alors aucun point appartenant strictement au quadrilatère `ABCD` est sur l'enveloppe convexe de `Points`

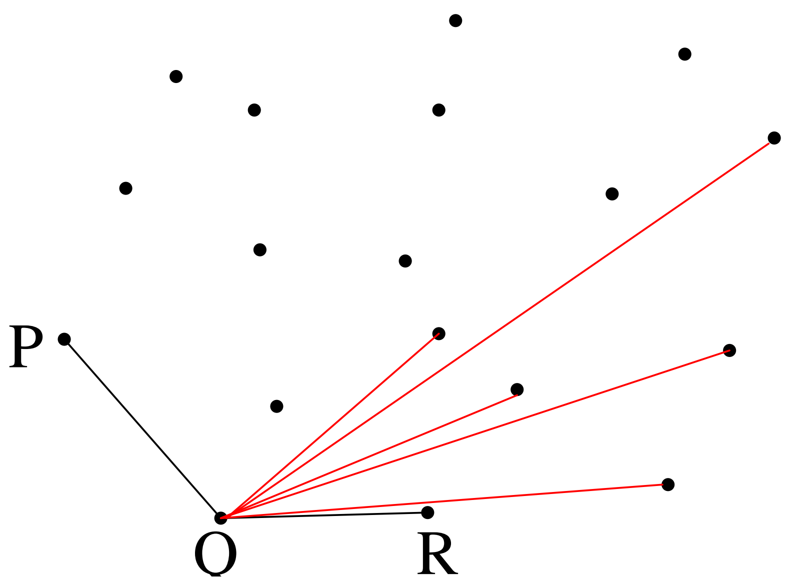


Enveloppe convexe : algorithmes

Notes pour révision : voir TME2

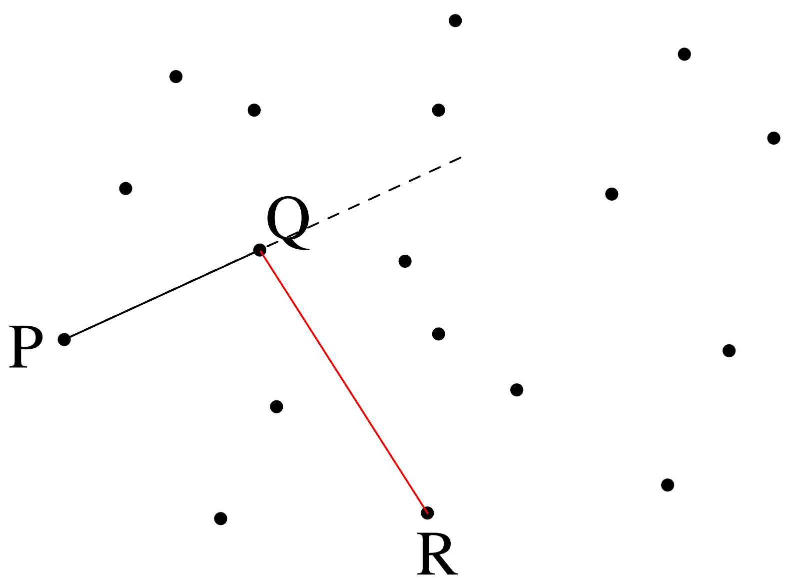
Algorithme Jarvis

IDÉE : si `PQ` est un côté de l'enveloppe convexe de `Points`, et `R` le point de `Points` tel que l'angle $(\overrightarrow{\mathtt{PQ}}, \overrightarrow{\mathtt{QR}})$ est minimum, alors `QR` est un côté de l'enveloppe convexe de `Points`



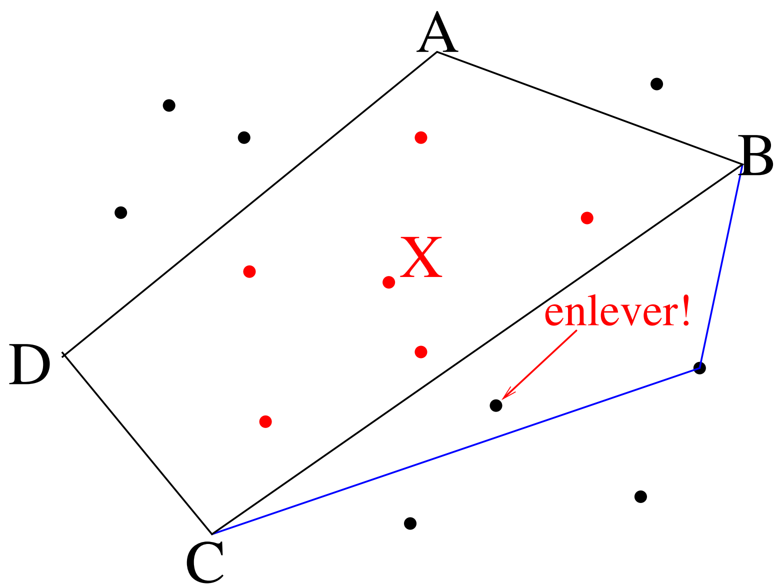
Algorithme Graham

IDÉE : si `P`, `Q` et `R` sont trois points de `Points` d'abscisse croissant, et `R` appartient au "mauvais côté" des demi-plans définis par `PQ`, alors `Q` n'est pas sur l'enveloppe convexe de `Points`



Algorithme QuickHull

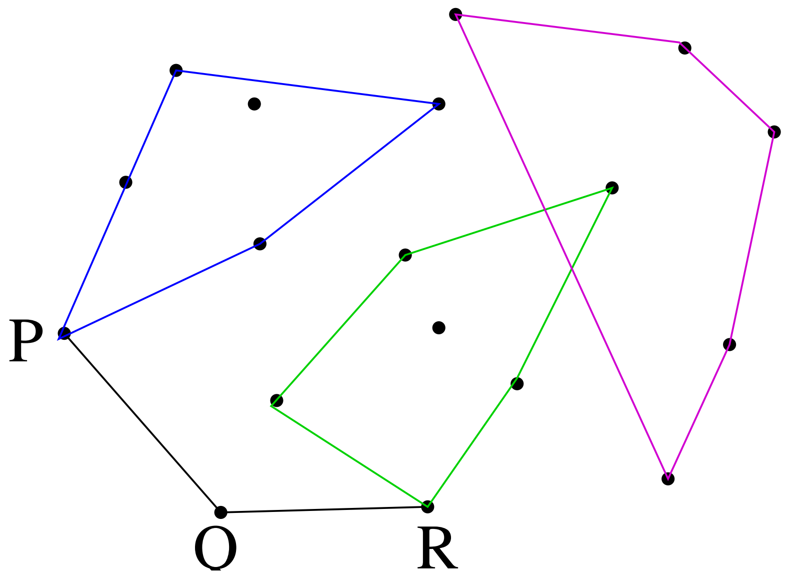
IDÉE : Akl-Toussaint est génial, comment l'étendre ?



Peut on faire mieux ?

Algorithme Chan

IDÉE : Jarvis + décomposition



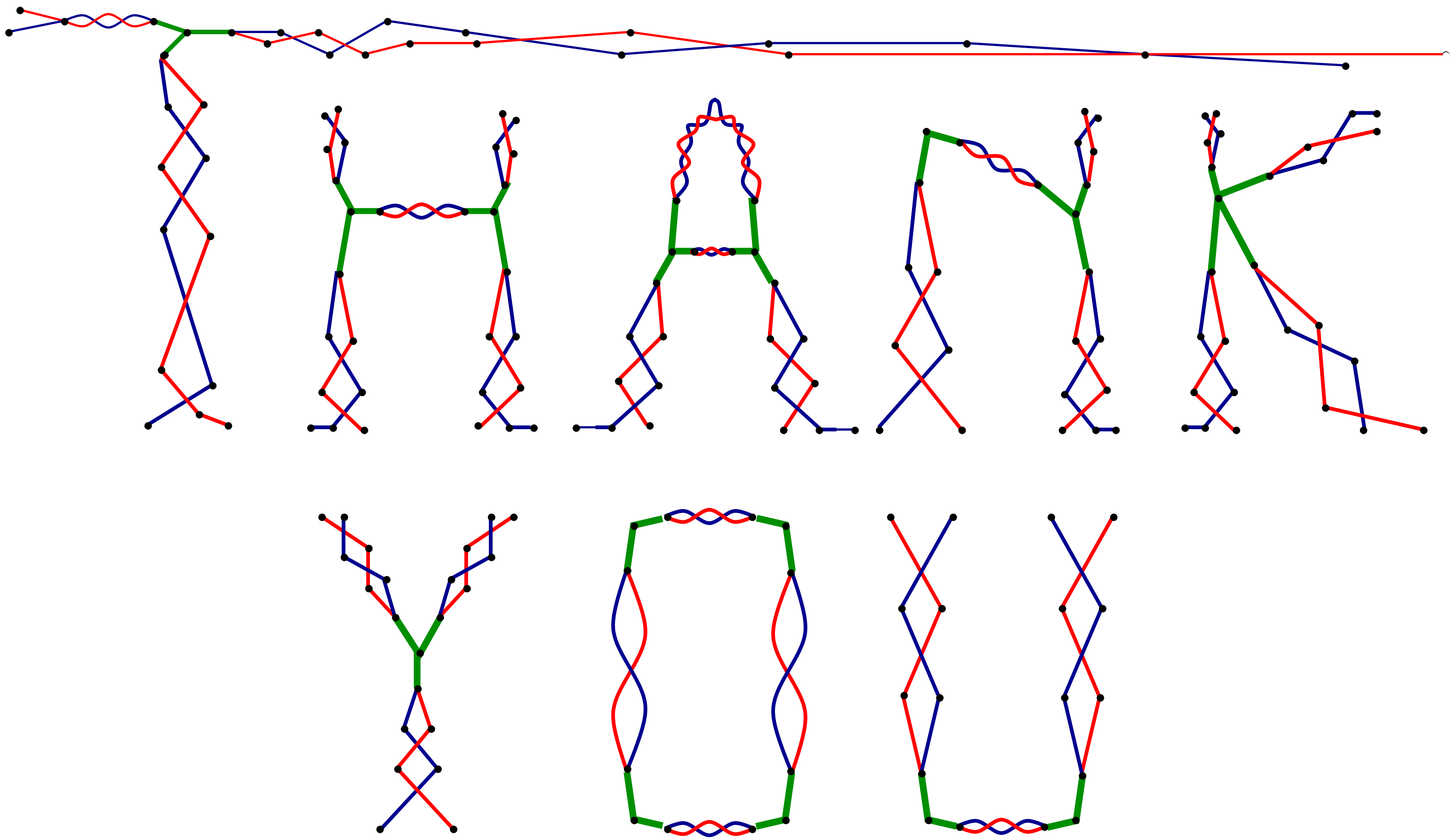
Conclusion, question

CONCLUSION :

- algorithme naïf en $O(n^3)$
- technique incrémental (Jarvis)
- technique de précalcul (pixel, Akl-Toussaint, QuickHull)
- technique d'élagage (Graham)
- technique ultime : décomposition (Chan)

QUESTION :

- implantation ? (voir TME)


THANK
YOU