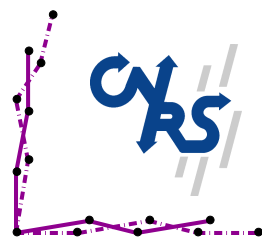


# Conception et Pratique de l'Algorithmique

<http://www-apr.lip6.fr/~buixuan/cpa2015>

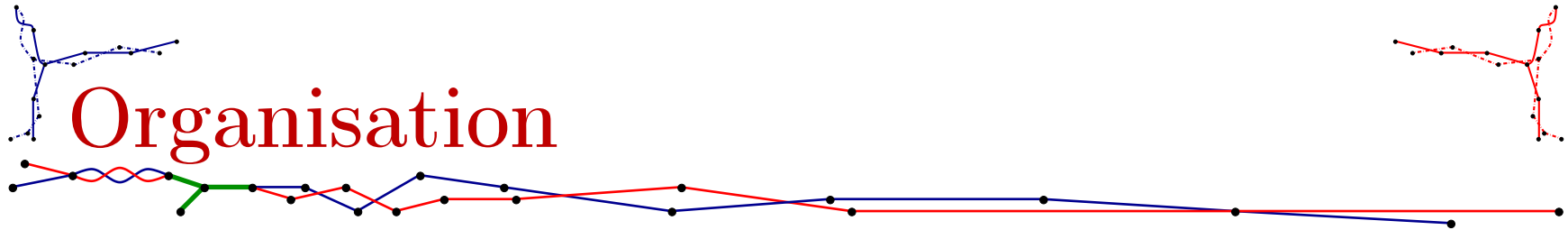
Binh-Minh Bui-Xuan



UPMC  
PARIS UNIVERSITAS

PARIS, Janvier 2015



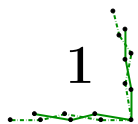
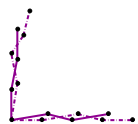


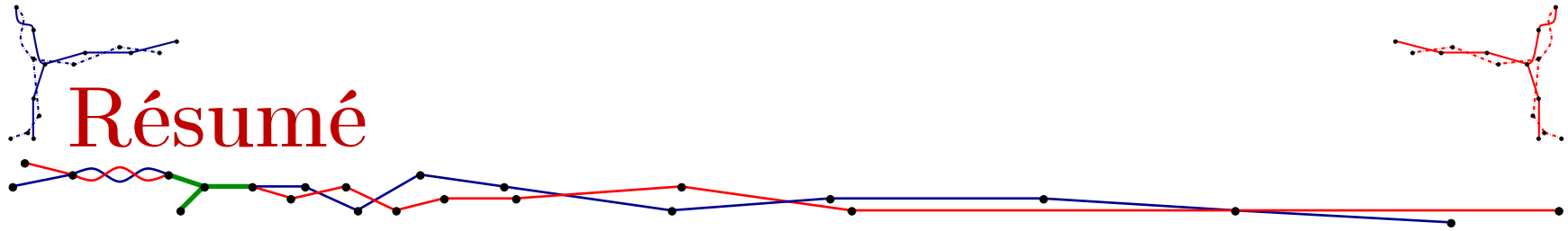
## EQUIPE PÉDAGOGIQUE :

- Binh-Minh Bui-Xuan (semaines 1-3, 10)
- Philippe Trébuchet (semaines 4-6)
- Christoph Dürr (semaines 7-9)
- Vincent Botbol (TME)

## CONTRÔLES :

- nombreux projets
- session 1 = projets + examen écrit semaine 18 Mai
- session 2 = examen écrit uniquement





## GÉOMÉTRIE ALGORITHMIQUE :

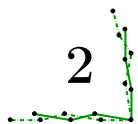
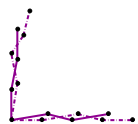
- collision d'objets, conteneur, diamètre
- implantation, évaluation

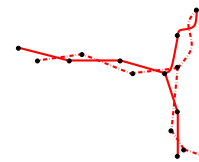
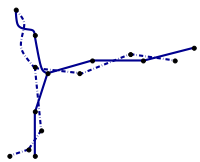
## ALGORITHME TEXTUEL :

- expression régulière, automate, déterminisation, optimisation
- implantation

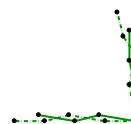
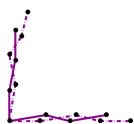
## ALGORITHME DE FLUX :

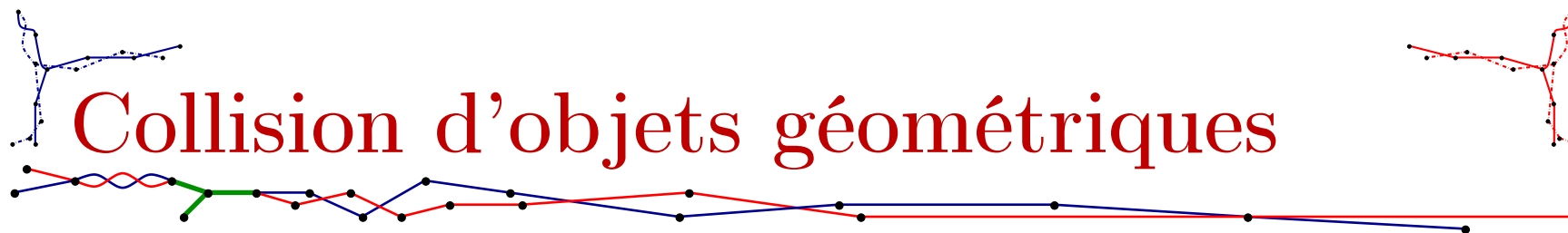
- complexité sous-linéaire
- problème des masses de données en *streaming*



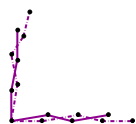
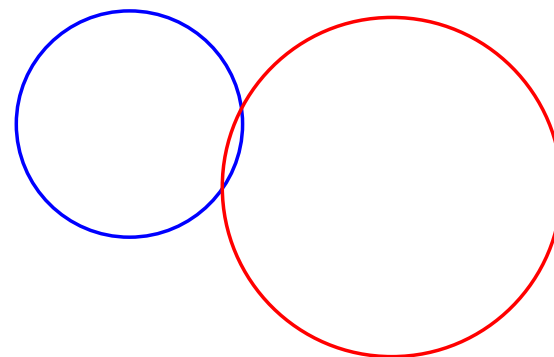
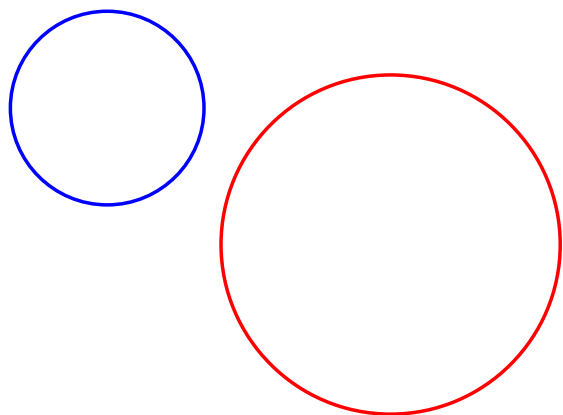


# Géométrie algorithmique





QUESTION : touché ?



# Collision d'objets géométriques

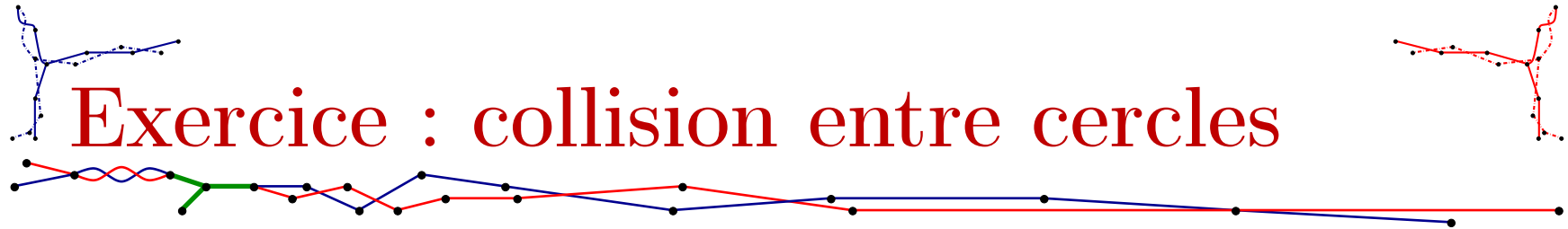
QUESTION : touché ?



# Collision d'objets géométriques

QUESTION : touché ?





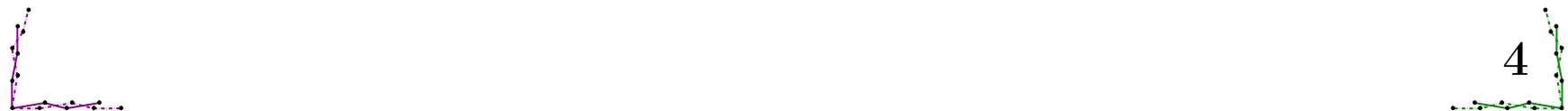
## Exercice : collision entre cercles

EXERCICE : Soient deux cercles  $c1$  et  $c2$  de rayons  $c1.radius$  et  $c2.radius$ , dont les coordonnées des centres sont  $(c1.x, c1.y)$  et  $(c2.x, c2.y)$ . Déterminer une condition nécessaire et suffisante pour que les deux cercles s'intersectent.

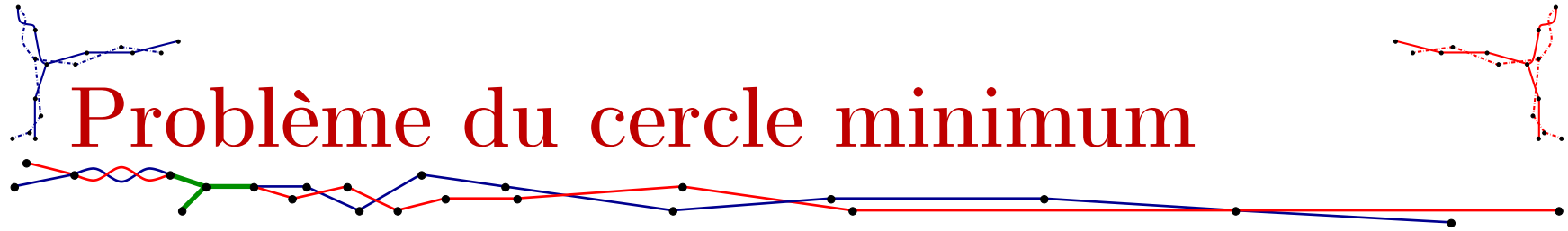
SUPPORT :

[http://www-apr.lip6.fr/~buixuan/files/RBB\\_collision\\_canevas.html](http://www-apr.lip6.fr/~buixuan/files/RBB_collision_canevas.html)

QUESTION : erreurs d'arrondi ?

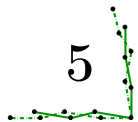
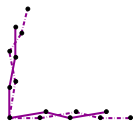


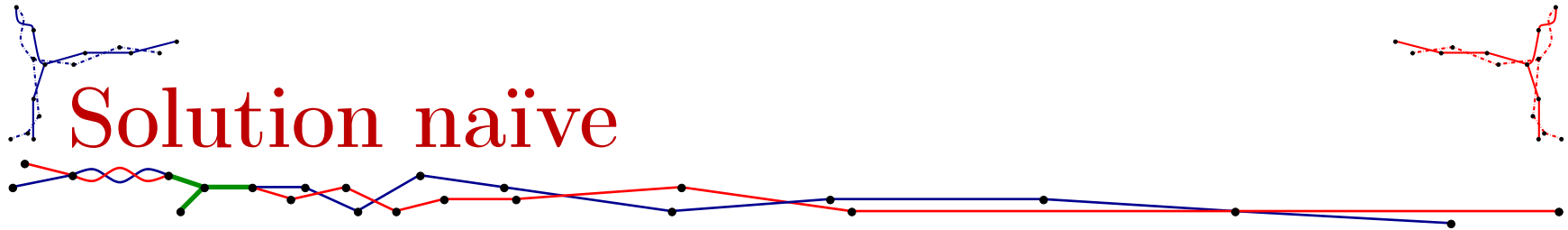




IN : Points, une liste de coordonnées de points en 2D

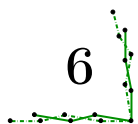
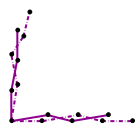
OUT : cercle couvrant tous les points de la liste, de rayon minimum

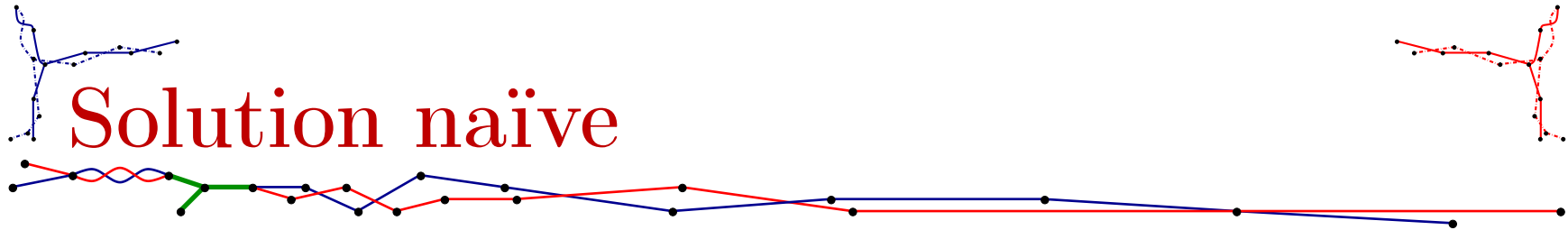




LEMME 1 : *si un cercle de diamètre égale à la distance de deux points de la liste couvre tout autre point de la liste, alors ce cercle est un cercle couvrant de rayon minimum.*

LEMME 2 : *en 2D, il existe un et un seul cercle passant par 3 points non-colinéaires.*

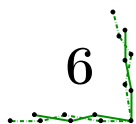
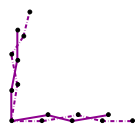


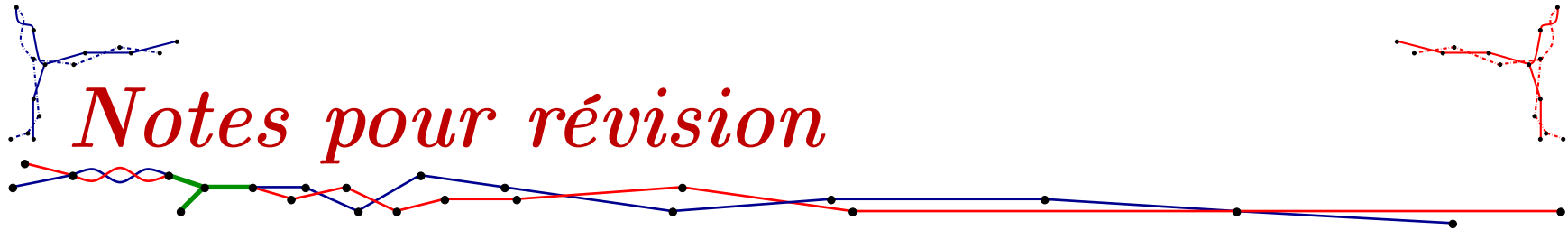


LEMME 1 : *si un cercle de diamètre égale à la distance de deux points de la liste couvre tout autre point de la liste, alors ce cercle est un cercle couvrant de rayon minimum.*

LEMME 2 : *en 2D, il existe un et un seul cercle passant par 3 points non-colinéaires.*

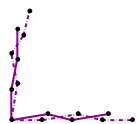
THÉORÈME : *le problème du cercle minimum peut être résolu en temps  $O(n^4)$ .*

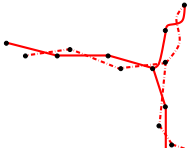
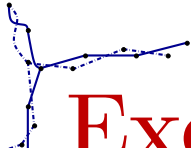




L'algorithme naïf en question :

1. pour tout  $p$  dans Points
2.   pour tout  $q$  dans Points
3.      $c \leftarrow$  cercle de centre  $\frac{p+q}{2}$  de diamètre  $|pq|$
4.     si  $c$  couvre tous les points de Points alors retourner  $c$
5. pour tout  $p$  dans Points
6.   pour tout  $q$  dans Points
7.     pour tout  $r$  dans Points
8.      $c \leftarrow$  cercle circonscrit de  $p$ ,  $q$  et  $r$
9.     si  $c$  couvre tous les points de Points alors retourner  $c$



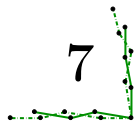
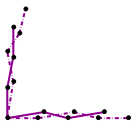


# Exercice : estimation du temps



QUESTION : un ordinateur de l'ordre du Giga-Hertz exécutant un algorithme en  $O(n^4)$ , avec  $n = 10000$ , quel est le temps d'exécution attendu (en ordre de grandeur) ?

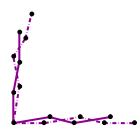
- algorithme en  $O(n^3)$  ?
- algorithme en  $O(n^2)$  ?
- algorithme en  $O(n)$  ?
- algorithme en  $O(\log n)$  ?

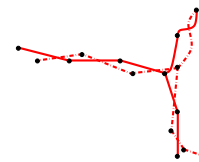
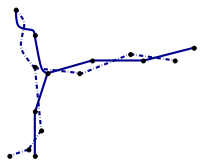




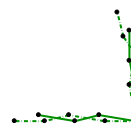
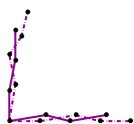
Qualité du résultat vs. temps d'exécution :

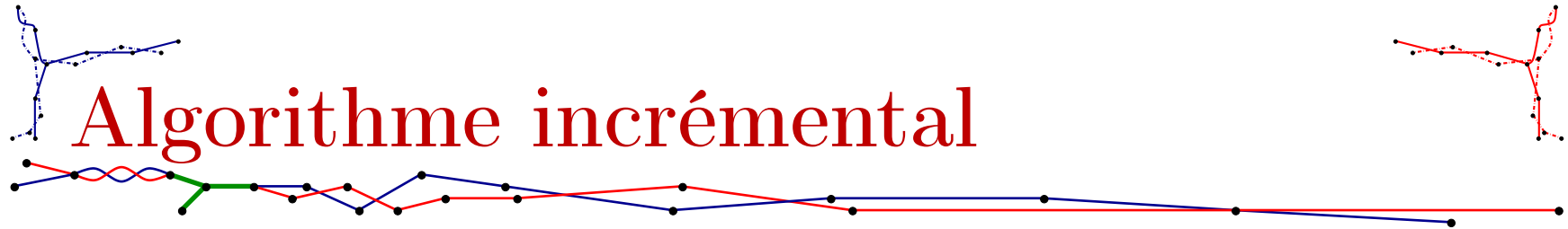
Qualité	Temps	Mixte
imagerie médicale	openGL	top-coder
appli. critiques	audio-visuel	optimisation





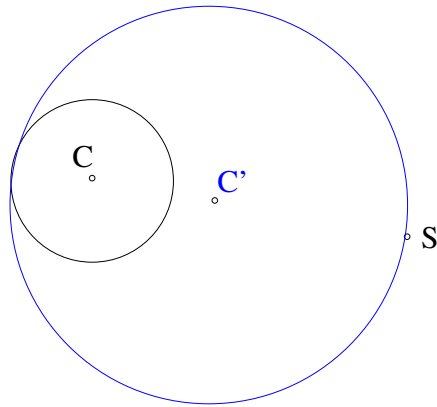
# Techniques d'approximation






# Algorithme incrémental


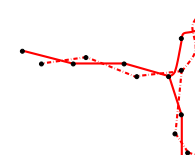
IDÉE : si un cercle ne couvre pas tous les points, on l'agrandit pour couvrir l'ancien cercle, plus au moins un nouveau point.



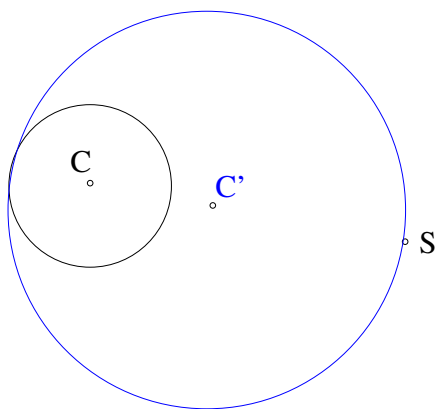




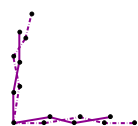
# Algorithme incrémental




IDÉE : si un cercle ne couvre pas tous les points, on l'agrandit pour couvrir l'ancien cercle, plus au moins un nouveau point.


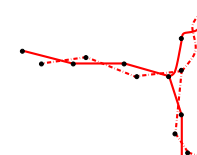


QUESTION : coordonnées de  $C'$  sachant  $C$ ,  $S$ , ancien rayon  $r$  ?





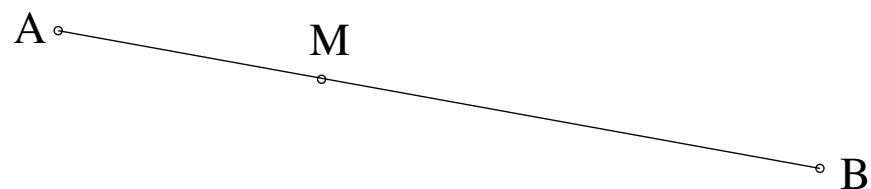
# Coordonnées barycentriques




FORMULE :  $M = \alpha \cdot A + \beta \cdot B$ , avec

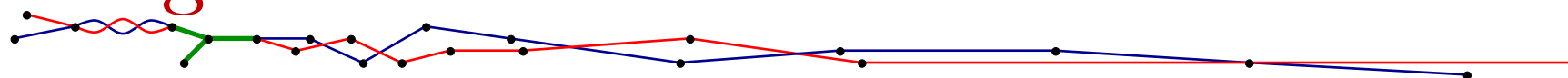
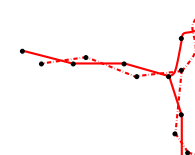
–  $\alpha = |MB|/|AB|$

–  $\beta = |MA|/|AB|$

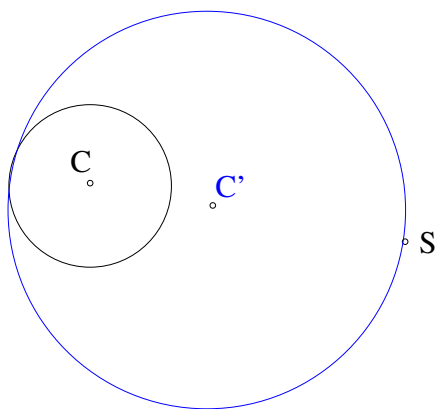




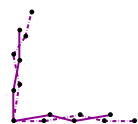
# Algorithme incrémental

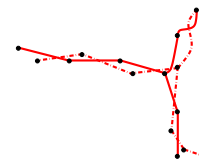
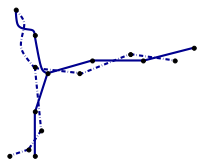


IDÉE : si un cercle ne couvre pas tous les points, on l'agrandit pour couvrir l'ancien cercle, plus au moins un nouveau point.

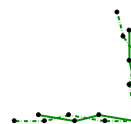
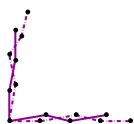



QUESTION : coordonnées de  $C'$  sachant  $C$ ,  $S$ , ancien rayon  $r$  ?




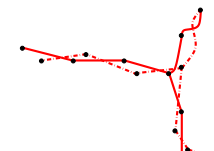


Peut on faire mieux ?

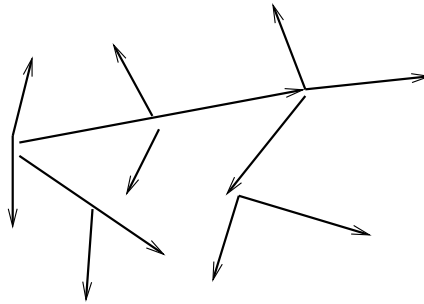




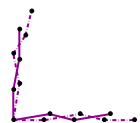
# Vue générale de l'algorithmique

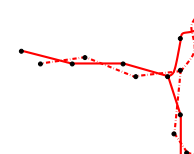



NAÏF : parcours exhaustif de l'espace de recherche




EXEMPLE : pour toute coordonnée possible du centre du cercle, pour toute valeur possible du rayon du cercle, vérifier si tous les points sont couverts.

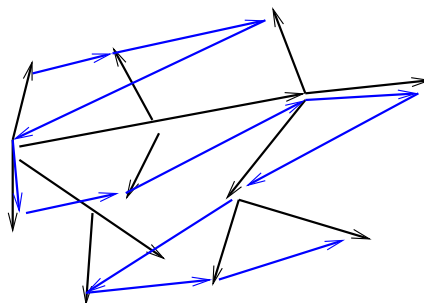




# Vue générale de l'algorithmique

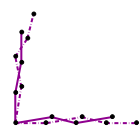


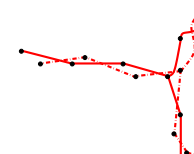

PARCOURS FUTÉ : réorganisation de l'espace de recherche




EXEMPLE 1 : voir partie “algorithmique du texte” semaines 4,5,6.

EXEMPLE 2 : pour toute coordonnée possible du centre *définie à partir de deux ou de trois points de la liste, soit  $\times \times \times$  la seule valeur intéressante du rayon*, vérifier si tous les points sont couverts.

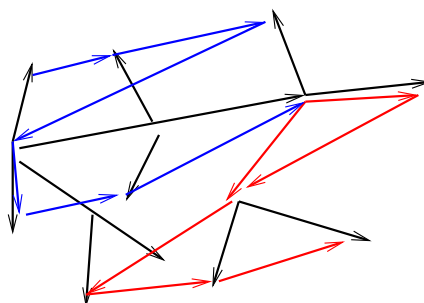




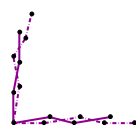
# Vue générale de l'algorithmique




PARCOURS EN AVEUGLE : réorganisation de l'espace de recherche +  
localisation

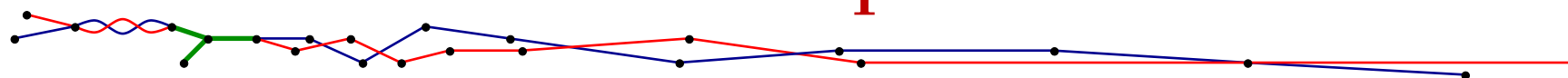
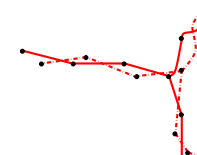


EXEMPLE 1 : voir partie “algorithmique sous-linéaire” semaines 7,8,9.  
EXEMPLE 2 : filtrer la donnée par un précalcul (cf. page suivante).

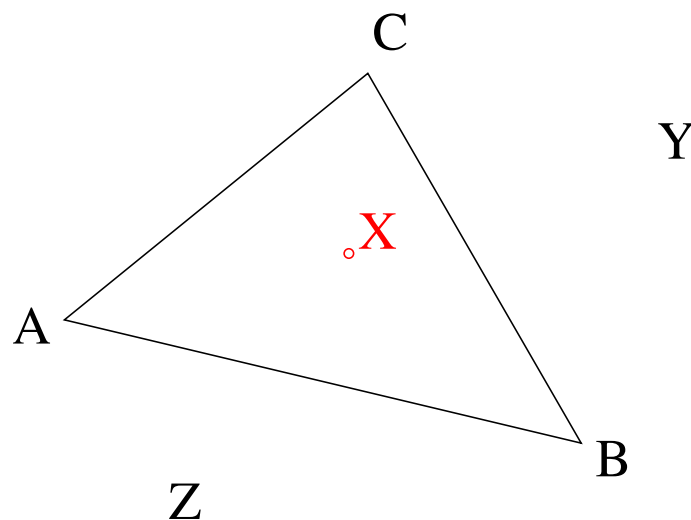




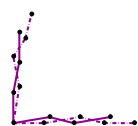
# Localisation avec le précalcul



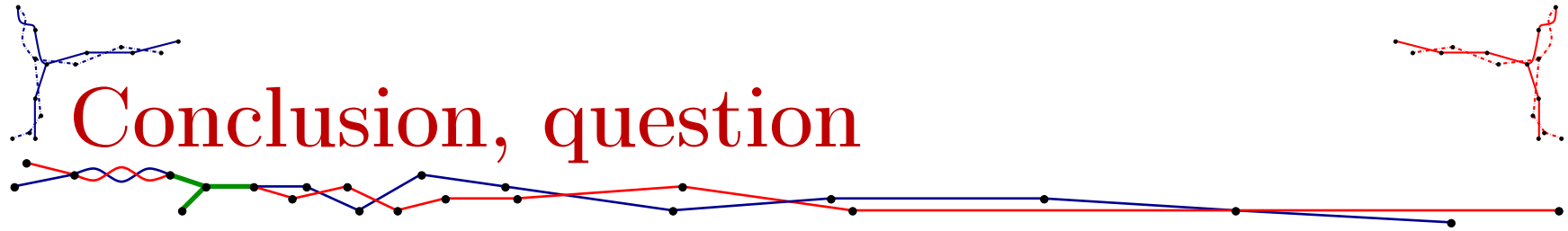
IDÉE : filtrer le “INPUT” pour écarter les zones de recherche inutiles



QUESTION :  $X$  est inutile, comment le détecter, numériquement ?







# Conclusion, question

CONCLUSION :

- algorithme naïf
- technique incrémental
- technique de filtrage (précalcul)

QUESTION :

- meilleur précalcul ? (voir TME pour une réponse)

