UNIVERSITÉ DE SHERBROOKE

IFT 436

Version 1.3
©2013
R. St-Denis

Algorithmes et structures de données
IFT 436

Richard St-Denis[1]

[1]Département d'informatique
Université de Sherbrooke

Automne 2017

UNIVERSITÉ DE SHERBROOKE

Chapitre 6
Quelques stratégies de conception

Première partie I

Objectifs du cours

Objectifs du cours

- Comprendre la stratégie *gloutonne* dans la conception d'algorithmes
- Comprendre la stratégie *inductive* dans la conception d'algorithmes
- Comprendre la stratégie *diviser-pour-régner* dans la conception d'algorithmes
- Comprendre la stratégie *couper-résoudre* dans la conception d'algorithmes
- Comprendre les algorithmes du calcul de l'enveloppe convexe
- Comprendre l'analyse de la complexité des algorithmes du calcul de l'enveloppe convexe

Objectifs du cours

- Comprendre la stratégie *gloutonne* dans la conception d'algorithmes
- Comprendre la stratégie *inductive* dans la conception d'algorithmes
- Comprendre la stratégie *diviser-pour-régner* dans la conception d'algorithmes
- Comprendre la stratégie *couper-résoudre* dans la conception d'algorithmes
- Comprendre les algorithmes du calcul de l'enveloppe convexe
- Comprendre l'analyse de la complexité des algorithmes du calcul de l'enveloppe convexe

Objectifs du cours

- Comprendre la stratégie *gloutonne* dans la conception d'algorithmes
- Comprendre la stratégie *inductive* dans la conception d'algorithmes
- Comprendre la stratégie *diviser-pour-régner* dans la conception d'algorithmes
- Comprendre la stratégie *couper-résoudre* dans la conception d'algorithmes
- Comprendre les algorithmes du calcul de l'enveloppe convexe
- Comprendre l'analyse de la complexité des algorithmes du calcul de l'enveloppe convexe

Objectifs du cours

- Comprendre la stratégie *gloutonne* dans la conception d'algorithmes
- Comprendre la stratégie *inductive* dans la conception d'algorithmes
- Comprendre la stratégie *diviser-pour-régner* dans la conception d'algorithmes
- Comprendre la stratégie *couper-résoudre* dans la conception d'algorithmes
- Comprendre les algorithmes du calcul de l'enveloppe convexe
- Comprendre l'analyse de la complexité des algorithmes du calcul de l'enveloppe convexe

Objectifs du cours

- Comprendre la stratégie *gloutonne* dans la conception d'algorithmes
- Comprendre la stratégie *inductive* dans la conception d'algorithmes
- Comprendre la stratégie *diviser-pour-régner* dans la conception d'algorithmes
- Comprendre la stratégie *couper-résoudre* dans la conception d'algorithmes
- Comprendre les algorithmes du calcul de l'enveloppe convexe
- Comprendre l'analyse de la complexité des algorithmes du calcul de l'enveloppe convexe

Objectifs du cours

- Comprendre la stratégie *gloutonne* dans la conception d'algorithmes
- Comprendre la stratégie *inductive* dans la conception d'algorithmes
- Comprendre la stratégie *diviser-pour-régner* dans la conception d'algorithmes
- Comprendre la stratégie *couper-résoudre* dans la conception d'algorithmes
- Comprendre les algorithmes du calcul de l'enveloppe convexe
- Comprendre l'analyse de la complexité des algorithmes du calcul de l'enveloppe convexe

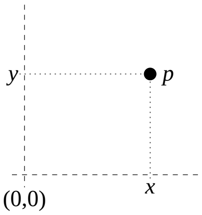
Deuxième partie II

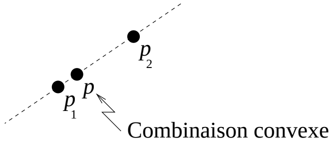
Contenu du cours

Plan du cours

1. Un peu de géométrie
   - Concepts de base
   - Produit croisé
   - Polygone
2. Quelques algorithmes pour le calcul de l'enveloppe convexe
   - Stratégie gloutonne
   - Stratégie inductive
   - Stratégie diviser-pour-régner
   - Stratégie couper-résoudre (incluant élaguer-chercher)
3. Explications additionnelles
   - Solutions d'équations avec le théorème maître
   - Solutions d'équations par expansion
   - Algorithme du calcul de la médiane
4. Rappels de résultats élémentaires
   - Séries géométriques
   - Logarithme

Point et combinaison convexe





Définition

Un point p est décrit par deux coordonnées cartésiennes $(x, y) \in \mathbb{R}^2$ par rapport à l'origine $(0, 0)$.

Définition

Si $p = \alpha p_1 + (1 - \alpha)p_2$, alors le point p est une combinaison convexe des deux points p_1 et p_2.

Segment et vecteur

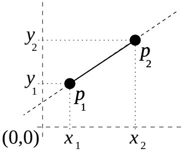
Définition

L'ensemble $\{p \mid (\exists \alpha \mid 0 \leq \alpha \leq 1 : p = \alpha p_1 + (1-\alpha)p_2)\}$ définit le segment de droite sous-tendu par les deux points p_1 et p_2, appelés extrémités, et noté $\overline{p_1 p_2}$.

Définition

Un vecteur est un segment de droite orienté, noté $\overrightarrow{p_1 p_2}$, de l'origine p_1 vers p_2 (si $p_1 = (0,0)$, le vecteur $\overrightarrow{p_1 p_2}$ est tout simplement noté $\overrightarrow{p_2}$).

Pente



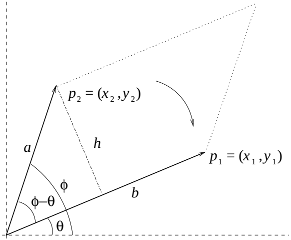
Notation

Les termes $x(p)$ et $y(p)$ désignent respectivement l'abscisse (coordonnée horizontale) et l'ordonnée (coordonnée verticale) du point p.

Définition

La pente d'une droite qui supporte le segment $\overline{p_1 p_2}$, notée $sl(p_1, p_2)$, est définie par la formule suivante : $sl(p_1, p_2) = (y(p_1) - y(p_2))/(x(p_1) - x(p_2)) = (y_1 - y_2)/(x_1 - x_2)$.

Aire d'un parallélogramme



Aire ($0 \le \phi - \theta \le \pi$)

$$\begin{aligned} A = b \times h &= b \times a \sin(\phi - \theta) \ge 0 \\ &= b \times a(\sin\phi\cos\theta - \cos\phi\sin\theta) \\ &= b\cos\theta \times a\sin\phi - b\sin\theta \times a\cos\phi \end{aligned}$$

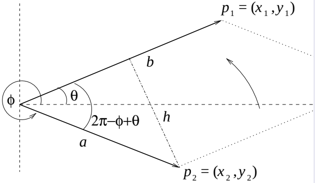
Récriture

$$A = x_1 y_2 - y_1 x_2 = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} = p_1 \times p_2 = -p_2 \times p_1 \ge 0$$

Remarque (pour p_1 par rapport à p_2)

Le produit croisé $p_1 \times p_2$ est positif et p_1 est dans le sens des aiguilles d'une montre par rapport à p_2.

Aire d'un parallélogramme



Aire ($\pi \leq \phi - \theta \leq 2\pi$)

$$\begin{aligned} A = b \times h &= b \times a \sin(2\pi + (\theta - \phi)) \\ &= b \times a \sin(\theta - \phi) \\ &= b \times (-a \sin(\phi - \theta)) \\ &= -(b \times a \sin(\phi - \theta)) \geq 0 \end{aligned}$$

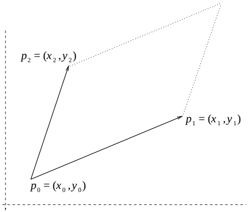
Récriture

$$A = -(b \times a \sin(\phi - \theta)) = -(p_1 \times p_2) \Rightarrow p_1 \times p_2 \leq 0$$

Remarque (pour p_1 par rapport à p_2)

Le produit croisé $p_1 \times p_2$ est négatif et p_1 est dans le sens inverse des aiguilles d'une montre par rapport à p_2.

Produit croisé si l'origine n'est pas $(0, 0)$



Aire (translation)

$$
\begin{aligned}
A &= (p_1 - p_0) \times (p_2 - p_0) \\
&= (x_1 - x_0)(y_2 - y_0) - (y_1 - y_0)(x_2 - x_0)
\end{aligned}
$$

Remarques (pour $\overrightarrow{p_0 p_1}$ par rapport à $\overrightarrow{p_0 p_2}$)

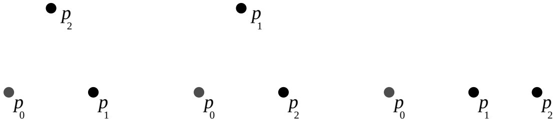
- Si A est positif, alors $\overrightarrow{p_0 p_1}$ est dans le sens des aiguilles d'une montre par rapport à $\overrightarrow{p_0 p_2}$.
- Si A est négatif, alors $\overrightarrow{p_0 p_1}$ est dans le sens inverse des aiguilles d'une montre par rapport à $\overrightarrow{p_0 p_2}$.

Ordre relatif de deux points p_1 et p_2 (l'origine est p_0)

Remarques (pour p_2 par rapport à p_1)

- Si $(p_2 - p_0) \times (p_1 - p_0) < 0$, alors p_2 est dans le sens inverse des aiguilles d'une montre par rapport à p_1 (virage à gauche).
- Si $(p_2 - p_0) \times (p_1 - p_0) > 0$, alors p_2 est dans le sens des aiguilles d'une montre par rapport à p_1 (virage à droite).
- Si $(p_2 - p_0) \times (p_1 - p_0) = 0$, alors p_0, p_1 et p_2 sont colinéaires.

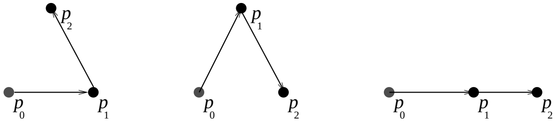
Ordre relatif de deux points p_1 et p_2 (l'origine est p_0)



Remarques (pour p_2 par rapport à p_1)

- Si $(p_2 - p_0) \times (p_1 - p_0) < 0$, alors p_2 est dans le sens inverse des aiguilles d'une montre par rapport à p_1 (virage à gauche).
- Si $(p_2 - p_0) \times (p_1 - p_0) > 0$, alors p_2 est dans le sens des aiguilles d'une montre par rapport à p_1 (virage à droite).
- Si $(p_2 - p_0) \times (p_1 - p_0) = 0$, alors p_0, p_1 et p_2 sont colinéaires.

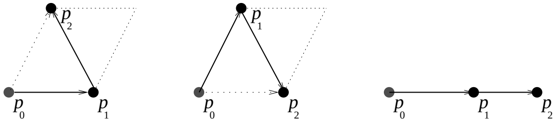
Ordre relatif de deux points p_1 et p_2 (l'origine est p_0)



Remarques (pour p_2 par rapport à p_1)

- Si $(p_2 - p_0) \times (p_1 - p_0) < 0$, alors p_2 est dans le sens inverse des aiguilles d'une montre par rapport à p_1 (virage à gauche).
- Si $(p_2 - p_0) \times (p_1 - p_0) > 0$, alors p_2 est dans le sens des aiguilles d'une montre par rapport à p_1 (virage à droite).
- Si $(p_2 - p_0) \times (p_1 - p_0) = 0$, alors p_0, p_1 et p_2 sont colinéaires.

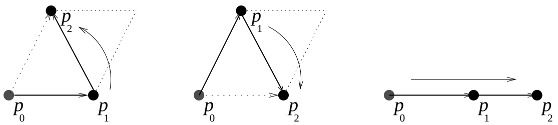
Ordre relatif de deux points p_1 et p_2 (l'origine est p_0)



Remarques (pour p_2 par rapport à p_1)

- Si $(p_2 - p_0) \times (p_1 - p_0) < 0$, alors p_2 est dans le sens inverse des aiguilles d'une montre par rapport à p_1 (virage à gauche).
- Si $(p_2 - p_0) \times (p_1 - p_0) > 0$, alors p_2 est dans le sens des aiguilles d'une montre par rapport à p_1 (virage à droite).
- Si $(p_2 - p_0) \times (p_1 - p_0) = 0$, alors p_0, p_1 et p_2 sont colinéaires.

Ordre relatif de deux points p_1 et p_2 (l'origine est p_0)



Remarques (pour p_2 par rapport à p_1)

- Si $(p_2 - p_0) \times (p_1 - p_0) < 0$, alors p_2 est dans le sens inverse des aiguilles d'une montre par rapport à p_1 (virage à gauche).
- Si $(p_2 - p_0) \times (p_1 - p_0) > 0$, alors p_2 est dans le sens des aiguilles d'une montre par rapport à p_1 (virage à droite).
- Si $(p_2 - p_0) \times (p_1 - p_0) = 0$, alors p_0, p_1 et p_2 sont colinéaires.

Définitions de polygone

Définition

Un polygone est un chemin fermé $p_0, \ldots, p_n$, avec $p_0 = p_n$, où les points $p_0, \ldots, p_{n-1}$ sont les sommets du polygone et les segments $\overline{p_0 p_1}$, ..., $\overline{p_{n-1} p_0}$ sont les arêtes.

Définition

Un polygone est dit simple si toute paire de segments $\overline{p_i p_{i \oplus 1}}$ et $\overline{p_j p_{j \oplus 1}}$, avec $i < j$, ont une intersection seulement si $i \oplus 1 = j$.

Définition

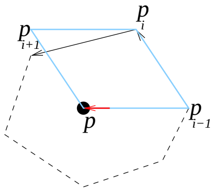
Un polygone simple est dit convexe si pour toute paire de points p_1 et p_2 situés à l'intérieur, tous les points du segment $\overline{p_1 p_2}$ sont aussi à l'intérieur.

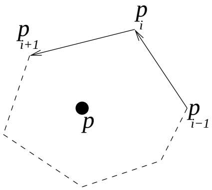
Le point p à l'intérieur d'un polygone convexe

Expression mathématique de p *à l'intérieur des deux segments*

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_i p_{i+1}}$).

Le point p à l'intérieur d'un polygone convexe

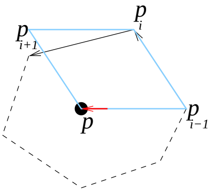


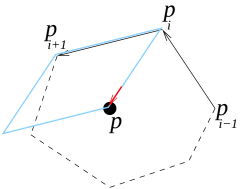


Expression mathématique de p *à l'intérieur des deux segments*

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_i p_{i+1}}$).

Le point p à l'intérieur d'un polygone convexe

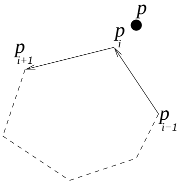


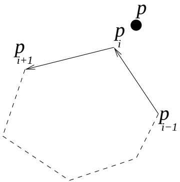


Expression mathématique de p *à l'intérieur des deux segments*

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_i p_{i+1}}$).

Le point p à l'extérieur d'un polygone convexe

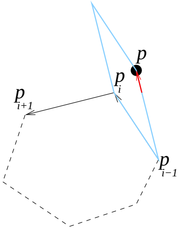


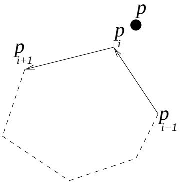


Expression mathématique de p *à l'extérieur des deux segments*

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_i p_{i+1}}$).

Le point p à l'extérieur d'un polygone convexe





Expression mathématique de p *à l'extérieur des deux segments*

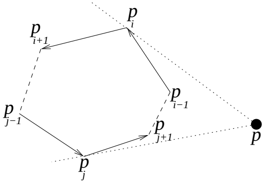
- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_i p_{i+1}}$).

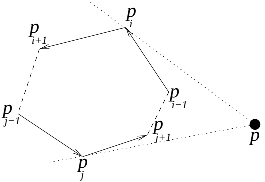
Le point p à l'extérieur d'un polygone convexe

Expression mathématique de p *à l'extérieur des deux segments*

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_i p_{i+1}}$).

Le segment $\overrightarrow{pp_i}$ tangent au polygone convexe

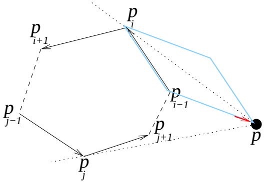


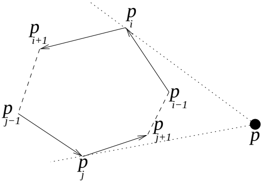


Tangente supérieure

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_i p_{i+1}}$). Ceci est le critère d'arrêt si le départ est depuis le point tangent inférieur.

Le segment $\overrightarrow{pp_i}$ tangent au polygone convexe

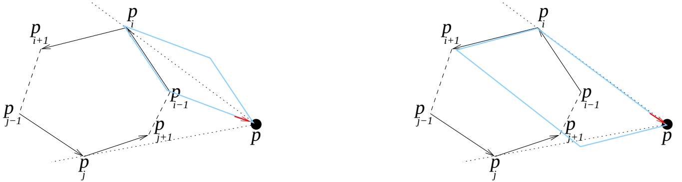




Tangente supérieure

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_i p_{i+1}}$). Ceci est le critère d'arrêt si le départ est depuis le point tangent inférieur.

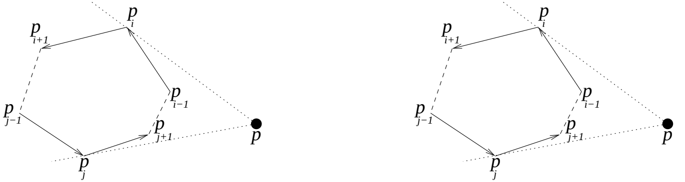
Le segment $\overrightarrow{pp_i}$ tangent au polygone convexe



Tangente supérieure

- Si $(p - p_{i-1}) \times (p_i - p_{i-1}) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_{i-1}p_i}$).
- Si $(p - p_i) \times (p_{i+1} - p_i) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_i p_{i+1}}$). Ceci est le critère d'arrêt si le départ est depuis le point tangent inférieur.

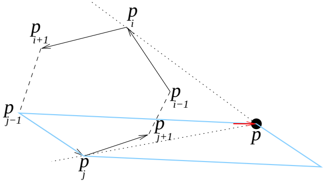
Le segment $\overrightarrow{pp_j}$ tangent au polygone convexe

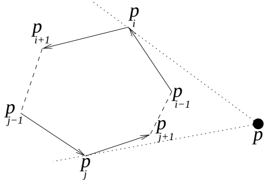


Tangente inférieure

- Si $({\color{red}p} - p_{j-1}) \times (p_j - p_{j-1}) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_{j-1}p_j}$). Ceci est le critère d'arrêt si le départ est depuis le point tangent supérieur.
- Si $({\color{red}p} - p_j) \times (p_{j+1} - p_j) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_jp_{j+1}}$).

Le segment $\overrightarrow{pp_j}$ tangent au polygone convexe

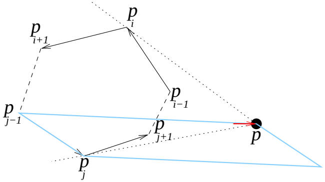


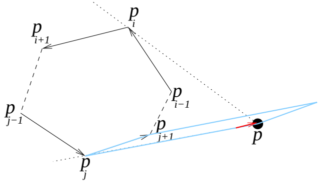


Tangente inférieure

- Si $(p - p_{j-1}) \times (p_j - p_{j-1}) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_{j-1}p_j}$). Ceci est le critère d'arrêt si le départ est depuis le point tangent supérieur.
- Si $(p - p_j) \times (p_{j+1} - p_j) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_j p_{j+1}}$).

Le segment $\overrightarrow{pp_j}$ tangent au polygone convexe





Tangente inférieure

- Si $(p - p_{j-1}) \times (p_j - p_{j-1}) < 0$, alors virage à gauche (p est à l'intérieur de $\overrightarrow{p_{j-1}p_j}$). Ceci est le critère d'arrêt si le départ est depuis le point tangent supérieur.
- Si $(p - p_j) \times (p_{j+1} - p_j) > 0$, alors virage à droite (p est à l'extérieur de $\overrightarrow{p_jp_{j+1}}$).

Enveloppe convexe

Définition

Un sous-ensemble S du plan est convexe si et seulement si pour toute paire de points p_1 et p_2 de S le segment de droite $\overline{p_1 p_2}$ est contenu dans S.

Définition

L'enveloppe convexe d'un ensemble de points S est le plus petit polygone convexe P pour lequel chaque point de S est soit sur la frontière de P ou à l'intérieur de P.

Stratégie gloutonne

Principe

À partir d'un problème P de taille n :

1. prendre un élément d'un ensemble (relatif au problème) ;
2. le traiter complètement (exactement une seule fois) ;
3. passer au prochain élément.

Remarque

La stratégie gloutonne suggère de visiter et de traiter chaque élément de l'ensemble associé au problème exactement une seule fois.

Stratégie gloutonne (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Remarque

Les données d'entrée ne correspondent pas nécessairement à l'ensemble considéré dans la stratégie gloutonne.

Illustration pour le calcul de l'enveloppe convexe

L'élément à considérer à chaque étape est

Stratégie gloutonne (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Remarque

Les données d'entrée ne correspondent pas nécessairement à l'ensemble considéré dans la stratégie gloutonne.

Illustration pour le calcul de l'enveloppe convexe

L'élément à considérer à chaque étape est un segment :

Stratégie gloutonne (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Remarque

Les données d'entrée ne correspondent pas nécessairement à l'ensemble considéré dans la stratégie gloutonne.

Illustration pour le calcul de l'enveloppe convexe

L'élément à considérer à chaque étape est un segment :

1. prendre un segment ;

Stratégie gloutonne (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Remarque

Les données d'entrée ne correspondent pas nécessairement à l'ensemble considéré dans la stratégie gloutonne.

Illustration pour le calcul de l'enveloppe convexe

L'élément à considérer à chaque étape est un segment :

1. prendre un segment ;
2. déterminer si le segment est une arête du polygone convexe ;

Stratégie gloutonne (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Remarque

Les données d'entrée ne correspondent pas nécessairement à l'ensemble considéré dans la stratégie gloutonne.

Illustration pour le calcul de l'enveloppe convexe

L'élément à considérer à chaque étape est un segment :

1. prendre un segment ;
2. déterminer si le segment est une arête du polygone convexe ;
3. recommencer la deuxième étape avec un autre segment.

Stratégie gloutonne (exemple)

Propriété d'un polygone convexe

Un segment est une arête si tous les points $p_1, \ldots, p_n$ sont orientés dans le même sens par rapport à ce segment.

Algorithme

$\{$Hypothèse : $n > 1\}$

for $i := 1$ to $n - 1$
    for $j := i + 1$ to n $\{$Prendre le segment $\overline{p_i p_j}\}$
        $s := 0$ $\{$Déterminer si le segment est une arête$\}$
        for $k := 1$ to n
            $a := (p_k - p_i) \times (p_j - p_i)$
            if $s = 0$ then $s := \mathrm{sign}(a)$
                else if $\ll s \neq \mathrm{sign}(a) \gg$ then break
        if $s = \mathrm{sign}(a)$ then p_i et p_j sommets du polygone

Stratégie gloutonne (exemple)

Analyse de la complexité algorithmique

- La boucle principale comporte $n - 1$ itérations.
- La boucle intermédiaire comporte $n - i$ itérations.
- La boucle la plus interne comporte n itérations.

Nombre total d'opérations dans la boucle la plus interne

nombre de segments $\times$ nombre d'opérations par segment

$(\sum_{k=1}^{n-1} k)n = \frac{n(n-1)}{2}n = \frac{n^3-n^2}{2}$

Conclusion

L'algorithme est en temps $O(n^3)$.

Stratégie gloutonnee

Spécialisation de stratégie gloutonne

Dans l'optique où le problème est formulé par rapport à l'atteinte d'un objectif, l'élément choisi à chaque étape est celui qui permet de s'approcher de l'objectif (généralement à partir d'un critère local).

Stratégie inductive

Principe

Soit $P(k)$ un problème de taille k (k élements comme donnée en entrée).

1. Le cas de base : il existe une solution facile pour $P(n_0)$.
2. Hypothèse d'induction : on suppose qu'il existe une solution pour $P(k)$, $k \geq n_0$.
3. Le pas d'induction : trouver une solution pour $P(k+1)$ à partir d'une solution pour $P(k)$ et de l'élément $k+1$.

Stratégie inductive

Formulation informelle (stratégie ascendante)

- Les cas de base : il existe des solutions pour quelques problèmes de taille minimale.
- Hypothèse d'induction : il existe des solutions pour des problèmes de taille inférieure ou égale à k.
- Le pas d'induction : construire une solution d'un problème de taille $k+1$ à partir de l'hypothèse d'induction.

Stratégie inductive

Formulation informelle (stratégie descendante)

- Les cas de base : il existe des solutions pour quelques problèmes de taille minimale.
- Hypothèse d'induction : il existe des solutions pour des problèmes de taille inférieure ou égale à k.
- Le pas d'induction : enlever un ou plusieurs objets du problème de taille $k+1$ pour appliquer l'hypothèse d'induction.

Stratégie inductive (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

Le problème $P(k)$ est

1. Le cas de base :
2. Hypothèse d'induction :
3. Le pas d'induction :

Stratégie inductive (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

Le problème $P(k)$ est le calcul de l'enveloppe convexe à partir de k points.

1. Le cas de base :
2. Hypothèse d'induction :
3. Le pas d'induction :

Stratégie inductive (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

Le problème $P(k)$ est le calcul de l'enveloppe convexe à partir de k points.

1. Le cas de base : $n_0 = 3$, car p_1, p_2, p_3 forment un triangle et donc un polygone convexe.
2. Hypothèse d'induction :
3. Le pas d'induction :

Stratégie inductive (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

Le problème $P(k)$ est le calcul de l'enveloppe convexe à partir de k points.

1. Le cas de base : $n_0 = 3$, car p_1, p_2, p_3 forment un triangle et donc un polygone convexe.
2. Hypothèse d'induction : on suppose connue la solution pour $P(k)$, c'est-à-dire $CH(k)$, $k \geq n_0$.
3. Le pas d'induction :

Stratégie inductive (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

Le problème $P(k)$ est le calcul de l'enveloppe convexe à partir de k points.

1. Le cas de base : $n_0 = 3$, car p_1, p_2, p_3 forment un triangle et donc un polygone convexe.
2. Hypothèse d'induction : on suppose connue la solution pour $P(k)$, c'est-à-dire $CH(k)$, $k \geq n_0$.
3. Le pas d'induction : l'enveloppe convexe $CH(k)$ de $p_1, \ldots, p_k$ est mise à jour par rapport à p_{k+1} pour obtenir l'enveloppe convexe $CH(k+1)$ de $p_1, \ldots, p_{k+1}$.

Stratégie inductive (exemple)

Possibilités à une étape donnée

À partir d'une numérotation des points de $CH(k)$ dans le sens inverse des aiguilles d'une montre :

Rappel

- Le point p_{k+1} est à l'intérieur de $CH(k)$.
  Il suffit de vérifier que p_{k+1} est à la gauche de toutes les arêtes de $CH(k)$.
- Le point p_{k+1} est à l'extérieur de $CH(k)$.
  Il suffit de trouver deux tangentes $\overline{p_{k+1}, p_i}$ et $\overline{p_{k+1}, p_j}$ à $CH(k)$ (i.e., $p_i, p_j \in CH(k)$) telles que :
  - si $\overrightarrow{p_{i-1}, p_i}$ (resp. $\overrightarrow{p_j, p_{j+1}}$) est dans le sens des aiguilles d'une montre par rapport à $\overrightarrow{p_{i-1}, p_{k+1}}$ (resp. $\overrightarrow{p_j, p_{k+1}}$), une boucle sur virages à droite, et $\overrightarrow{p_i, p_{i+1}}$ (resp. $\overrightarrow{p_{j-1}, p_j}$) est dans le sens inverse des aiguilles d'une montre par rapport à $\overrightarrow{p_i, p_{k+1}}$ (resp. $\overrightarrow{p_{j-1}, p_{k+1}}$), premier virage à gauche, alors p_i (resp. p_j) est le point tangent supérieur (resp. inférieur).

Stratégie inductive (exemple)

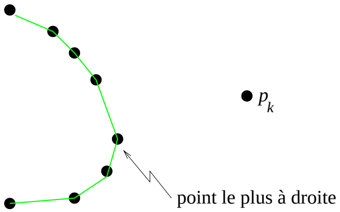
Élimination d'une possibilité

Pour éviter de vérifier si un point est à l'intérieur de $CH(k)$, il suffit de trier tous les points par ordre croissant par rapport à la coordonnée en abscisse. Ainsi tous les points non examinés seront toujours à l'extérieur de $CH(k)$ à la condition de les examiner en ordre.

Stratégie inductive (exemple)

Éliminer des points dans le triangle

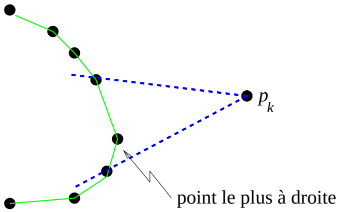
- le point p_k (le nouveau point à traiter)
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens inverse des aiguilles d'une montre
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens des aiguilles d'une montre



Stratégie inductive (exemple)

Éliminer des points dans le triangle

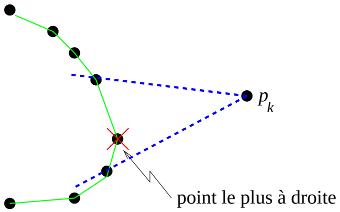
- le point p_k (le nouveau point à traiter)
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens inverse des aiguilles d'une montre
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens des aiguilles d'une montre



Stratégie inductive (exemple)

Éliminer des points dans le triangle

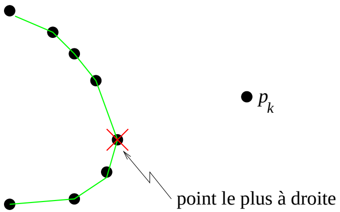
- le point p_k (le nouveau point à traiter)
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens inverse des aiguilles d'une montre
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens des aiguilles d'une montre



Stratégie inductive (exemple)

Éliminer des points dans le triangle

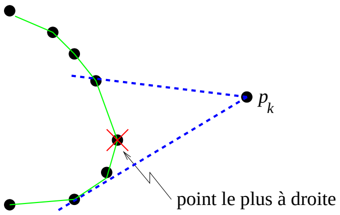
- le point p_k (le nouveau point à traiter)
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens inverse des aiguilles d'une montre
- le point suivant dans le sens des aiguilles d'une montre



Stratégie inductive (exemple)

Éliminer des points dans le triangle

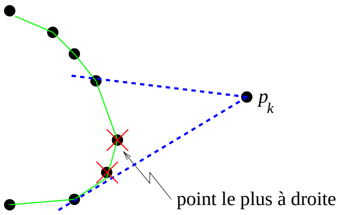
- le point p_k (le nouveau point à traiter)
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens inverse des aiguilles d'une montre
- le point suivant dans le sens des aiguilles d'une montre



Stratégie inductive (exemple)

Éliminer des points dans le triangle

- le point p_k (le nouveau point à traiter)
- le point suivant le point le plus à droite de $CH(k-1)$ dans le sens inverse des aiguilles d'une montre
- le point suivant dans le sens des aiguilles d'une montre



Stratégie inductive (exemple)

Algorithme – élimination des points à l'intérieur de triangles

{Hypothèse : $n > 2$}

$Sort(\langle p_1, p_2 \ldots, p_n \rangle)$
$CH := triangle(p_1, p_2, p_3)$

for $k := 4$ **to** n **do**
    let $CH = \langle q_1, \ldots, q_l \rangle$ **and** $q_i = RightmostVertex(CH)$
    $cw := q_{i \ominus 1}$ $ccw := q_{i \oplus 1}$
    $t_1 := (q_i - p_k) \times (ccw - p_k)$

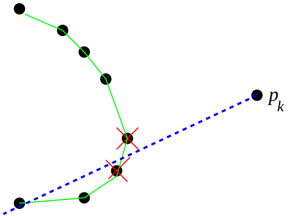
    while $cw \neq ccw$ **do**   {Calibrage}
        $t_2 := (q_i - p_k) \times (cw - p_k)$
        if $t_1 \times t_2 \geq 0$ **then break**
        $Remove(q_i)$ $q_i := cw$ $cw := q_{i \ominus 1}$

    if $cw = ccw$ **then** $cw := q_i$
    else $\cdots$

Stratégie inductive (exemple)

Calcul des tangentes

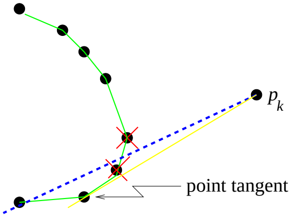
- Le point tangent inférieur (ou supérieur) est trouvé.
- Une recherche du point tangent supérieur (ou inférieur) est requise.



Stratégie inductive (exemple)

Calcul des tangentes

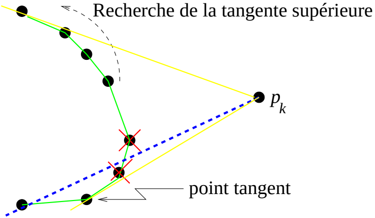
- Le point tangent inférieur (ou supérieur) est trouvé.
- Une recherche du point tangent supérieur (ou inférieur) est requise.



Stratégie inductive (exemple)

Calcul des tangentes

- Le point tangent inférieur (ou supérieur) est trouvé.
- Une recherche du point tangent supérieur (ou inférieur) est requise.



Stratégie inductive (exemple)

Algorithme – calcul des tangentes

⋮

for $k := 4$ **to** n **do**

⋮

    if $t_1 > 0$ **then**
        $ccw := q_i$ {Point tangent supérieur}
        $q_j := cw$ {Recherche du point tangent inférieur}
        do $t_2 := (p_k - q_{j\ominus 1}) \times (q_j - q_{j\ominus 1})$
            if $t_2 > 0$ **then** $j := j \ominus 1$
        while $t_2 > 0$ $cw := q_j$
    elseif $t_2 < 0$ **then**
        $cw := q_i$ {Point tangent inférieur}
        $q_i := ccw$ {Recherche du point tangent supérieur}
        do $t_1 := (p_k - q_i) \times (q_{i\oplus 1} - q_i)$
            if $t_1 > 0$ **then** $i := i \oplus 1$
        while $t_1 > 0$ $ccw := q_i$
    $Expand(CH, p_k, cw, ccw)$

Stratégie inductive (exemple)

Analyse de la complexité algorithmique

- Le tri à la ligne 1 est en temps $O(n \lg n)$.
- Pour un point donné p_k, le traitement dans le corps de la boucle principale consiste à éliminer des points.
- Comme il est impossible d'éliminer (et d'ajouter) plus de n points, les lignes 3 à 25 sont en temps O(n).

Conclusion

L'algorithme est en temps $O(n \lg n)$.

Stratégie diviser-pour-régner

Principe

À partir d'un problème P de taille n :

1. diviser le problème P en sous-problèmes disjoints $P_1, \ldots, P_l$;
2. de façon récursive, résoudre séparément $P_1, \ldots, P_l$ pour obtenir les solutions partielles $S_1, \ldots, S_l$;
3. combiner $S_1, \ldots, S_l$ pour obtenir une solution S de P.

Remarque

Si n_k désigne la taille du problème P_k $(1 \leq k \leq l)$, alors $\sum_{k=1}^{l} n_k = n$.

Stratégie diviser-pour-régner (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

À partir du problème P de taille n :

Stratégie diviser-pour-régner (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

À partir du problème P de taille n :

1. diviser S en $S_1 = \{p_1, \ldots, p_{\lfloor \frac{n}{2} \rfloor}\}$ et $S_2 = \{p_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, p_n\}$;

Stratégie diviser-pour-régner (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

À partir du problème P de taille n :

1. diviser S en $S_1 = \{p_1, \ldots, p_{\lfloor \frac{n}{2} \rfloor}\}$ et $S_2 = \{p_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, p_n\}$;
2. calculer récursivement l'enveloppe convexe de S_1, $CH(S_1)$, puis celle de S_2, $CH(S_2)$;

Stratégie diviser-pour-régner (exemple)

Donnée en entrée pour le calcul de l'enveloppe convexe

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe convexe

À partir du problème P de taille n :

1. diviser S en $S_1 = \{p_1, \ldots, p_{\lfloor \frac{n}{2} \rfloor}\}$ et $S_2 = \{p_{\lfloor \frac{n}{2} \rfloor + 1}, \ldots, p_n\}$;
2. calculer récursivement l'enveloppe convexe de S_1, $CH(S_1)$, puis celle de S_2, $CH(S_2)$;
3. combiner $CH(S_1)$ et $CH(S_2)$ pour obtenir l'enveloppe convexe $CH(S)$ de S.

Stratégie diviser-pour-régner (exemple)

Algorithme

$\{$Hypothèse : $y(p_i) < y(p_j) \iff i < j,\ 1 \leq i, j \leq n\}$

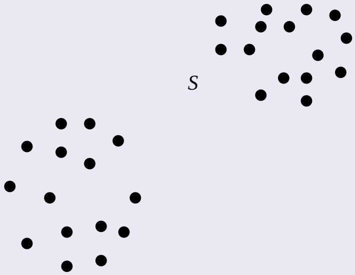
1. `function` $CH(S)$
2. `if size`$(S) = 1$ `then` $CH(S) := S$ {Cas limite}
3. $S_1 := \{p_1, \ldots, p_{\lfloor \frac{\texttt{size}(S)}{2} \rfloor}\}$ {Diviser}
4. $S_2 := \{p_{\lfloor \frac{\texttt{size}(S)}{2} \rfloor + 1}, \ldots, p_{\texttt{size}(S)}\}$
5. $ch_1 := CH(S_1)$ $ch_2 := CH(S_2)$ {Résoudre}
6. `return` $T(ch_1, ch_2)$ {Combiner}

Remarque

La difficulté réside dans l'agencement des deux enveloppes convexes de façon à en obtenir une plus grande.

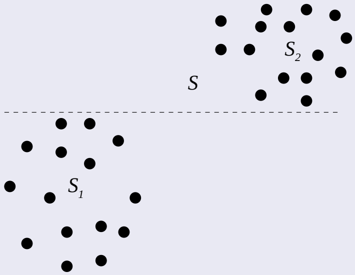
Stratégie diviser-pour-régner (combiner deux enveloppes convexes)

Illustration de la solution



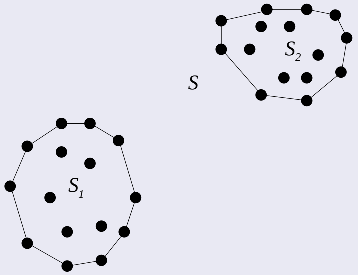
Stratégie diviser-pour-régner (combiner deux enveloppes convexes)

Illustration de la solution



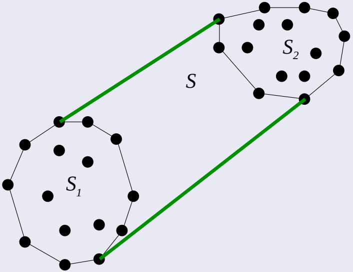
Stratégie diviser-pour-régner (combiner deux enveloppes convexes)

Illustration de la solution



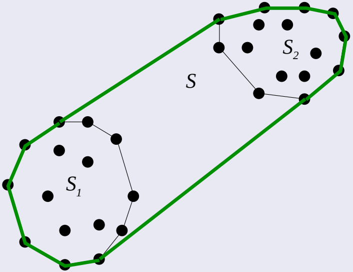
Stratégie diviser-pour-régner (combiner deux enveloppes convexes)

Illustration de la solution

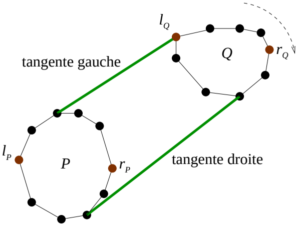


Stratégie diviser-pour-régner (combiner deux enveloppes convexes)

Illustration de la solution



Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



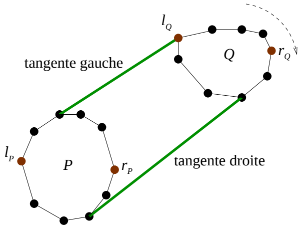
Éléments pour le calcul des tangentes

- $P = \{p_1, \ldots, p_l\}$, les sommets d'un polygone
- $Q = \{q_1, \ldots, q_m\}$, les sommets de l'autre polygone

Points particuliers

- l_P (resp. r_P) est le point le plus à gauche (resp. droite), $x(l_P) = \min_k x(p_k)$ (resp. $x(r_P) = \max_k x(p_k)$).
- l_Q (resp. r_Q) est le point le plus à gauche (resp. droite), $x(l_Q) = \min_k x(q_k)$ (resp. $x(r_Q) = \max_k x(q_k)$).

Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



Hypothèses

- $y(p_i) < y(q_j)$, $1 \leq i \leq l$ et $1 \leq j \leq m$ $(P \cap Q = \emptyset)$
- $p_1 = r_P$ et $q_1 = r_Q$
- Sens des aiguilles d'une montre

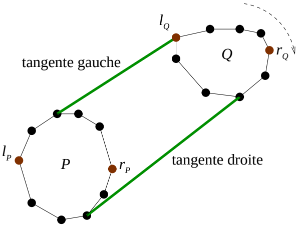
Un des cas pour le calcul de la tangente droite

$y(r_P) < y(r_Q)$ et $x(r_P) < x(r_Q)$

Idée : calcul de pentes

Pour une droite qui supporte le segment $\overline{uv}$,
$sl(u, v) = (y(u) - y(v))/(x(u) - x(v))$.

Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



But

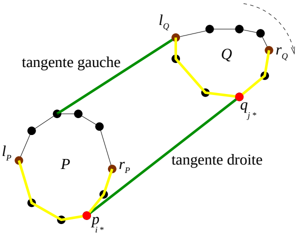
Trouver le sommet p_{i^*} de P et le sommet q_{j^*} de Q, les extrémités de la tangente droite, où

- $1 \leq i^* \leq \text{indice}(l_P)$;
- $1 \leq j^* \leq \text{indice}(l_Q)$.

Pentes de quelques droites

- $\alpha_{i,i\oplus 1} := sl(p_i, p_{i\oplus 1})$, pente d'une arête de P
- $\beta_{j,j\oplus 1} := sl(q_j, q_{j\oplus 1})$, pente d'une arête de Q
- $\gamma_{i,j} := sl(p_i, q_j)$, interprétation d'une tangente potentielle

Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



But

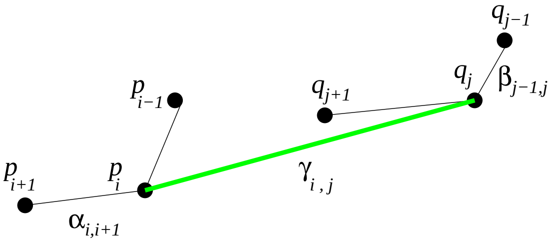
Trouver le sommet p_{i^*} de P et le sommet q_{j^*} de Q, les extrémités de la tangente droite, où

- $1 \leq i^* \leq \text{indice}(l_P)$;
- $1 \leq j^* \leq \text{indice}(l_Q)$.

Pentes de quelques droites

- $\alpha_{i,i\oplus 1} := sl(p_i, p_{i\oplus 1})$, pente d'une arête de P
- $\beta_{j,j\oplus 1} := sl(q_j, q_{j\oplus 1})$, pente d'une arête de Q
- $\gamma_{i,j} := sl(p_i, q_j)$, interprétation d'une tangente potentielle

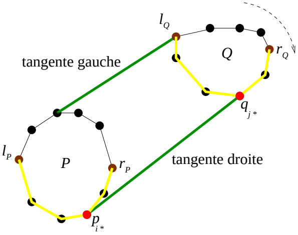
Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



Pentes de quelques droites

- $\alpha_{i,i\oplus 1} := sl(p_i, p_{i\oplus 1})$, pente d'une arête de P
- $\beta_{j,j\oplus 1} := sl(q_j, q_{j\oplus 1})$, pente d'une arête de Q
- $\gamma_{i,j} := sl(p_i, q_j)$, interprétation d'une tangente potentielle

Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



Pentes

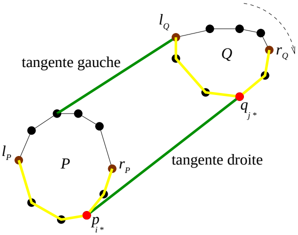
- $\alpha_{i,i\oplus 1} := sl(p_i, p_{i\oplus 1})$
- $\beta_{j,j\oplus 1} := sl(q_j, q_{j\oplus 1})$
- $\gamma_{i,j} := sl(p_i, q_j)$

Remarques

- $\alpha_{1,2}, \alpha_{2,3}, \ldots$ est une suite monotone strictement décroissante (pas de sommets colinéaires) et
- $\beta_{1,2}, \beta_{2,3}, \ldots$ est aussi une suite monotone strictement décroissante (pas de sommets colinéaires),

car P et Q sont des polygones convexes.

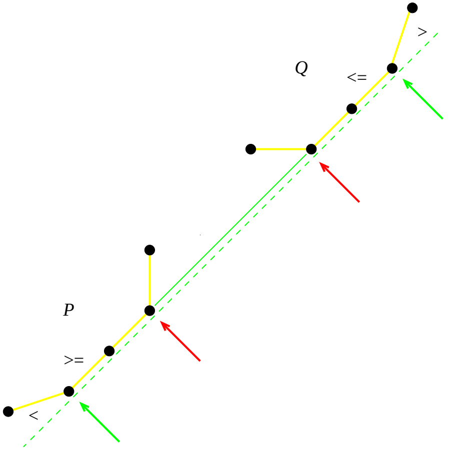
Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



Caractérisation de p_{j^*} et q_{j^*}

- $i^* > 1 \Rightarrow \alpha_{i^*,i^*\oplus 1} < \gamma_{i^*,j^*} \leq \alpha_{i^*\ominus 1,i^*}$
- $i^* = 1 \Rightarrow \alpha_{1,2} < \gamma_{1,j^*}$ (car 1-1 = 0 ; pas au-deça de 1)
- $j^* > 1 \Rightarrow \beta_{j^*,j^*\oplus 1} \leq \gamma_{i^*,j^*} < \beta_{j^*\ominus 1,j^*}$
- $j^* = 1 \Rightarrow \beta_{1,2} \leq \gamma_{i^*,1}$ (car 1-1 = 0 ; pas au-deça de 1)

Stratégie diviser-pour-régner (combiner deux enveloppes convexes)



Stratégie diviser-pour-régner (exemple)

Hypothèses

Les pentes $\alpha_{i,j}$ et $\beta_{i,j}$ sont déjà calculées.

Algorithme pour un des cas du calcul des extrémités de la tangente droite

1. $i := 1 \quad j := 1$
2. `loop`
3.   `loop`
4. $\gamma_{i,j} := y(p_i) - y(q_j))/(x(p_i) - x(q_j))$
5.   `if` $\alpha_{i,i\oplus 1} \geq \gamma_{i,j}$ `then` $i := i \oplus 1$ `else break`
6.   `if` $\beta_{j,j\oplus 1} > \gamma_{i,j}$ `then` $j := j \oplus 1$ `else break`
7. $i^* := i \quad j^* := j$

Stratégie diviser-pour-régner (exemple)

Analyse de la complexité du calcul des tangentes

- Le calcul de la tangente droite est en temps $O(i^* + j^*)$.
- Le calcul de la tangente gauche est en temps $O(i^{**} + j^{**})$.

Conclusion

L'algorithme est en temps $O(l + m) = O(n)$.

Remarque

La structure de données utilisée pour représenter un polygone convexe est une liste circulaire de ses sommets. Ainsi la construction de $CH(P, Q)$ à partir de $CH(P)$ et $CH(Q)$ est simplement faite par une mise à jour de deux pointeurs. Lesquels ?

Stratégie diviser-pour-régner (exemple)

Analyse de la complexité du calcul de l'enveloppe convexe

- Soit $CH(n)$ le nombre d'opérations requises pour calculer l'enveloppe convexe de n points.
- Soit $T(n)$ le nombre d'opérations requises pour calculer les tangentes.

Équation de récurrence

$$CH(n) = 2CH(\tfrac{n}{2}) + O(n)$$

À l'aide du cas 2 d'une variante du théorème *maître*

L'algorithme est en temps $O(n \lg n)$.

Explication

Stratégie couper-résoudre

Principe

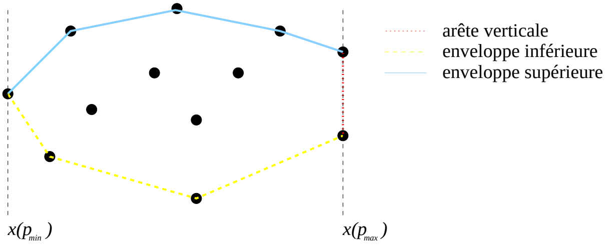
À partir d'un problème P de taille n :

1. déterminer la façon de couper P en sous-problèmes disjointes $P_1, \ldots, P_l$ et de combiner leur solution (sans nécessairement les connaître) ;
2. considérer les sous-problèmes disjoints $P_1, \ldots, P_l$ de P ;
3. de façon récursive, résoudre séparément $P_1, \ldots, P_l$ pour obtenir les solutions partielles $S_1, \ldots, S_l$.

Remarque

Si n_k désigne la taille du problème P_k $(1 \leq k \leq l)$, alors $\sum_{k=1}^{l} n_k = n$.

Stratégie couper-résoudre (enveloppe convexe)



Parties de l'enveloppe convexe

L'algorithme calcule l'enveloppe supérieure et l'enveloppe inférieure de l'enveloppe convexe.

Remarque

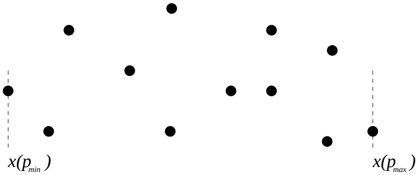
Seul le calcul de l'enveloppe supérieure est présenté.

Stratégie couper-résoudre (enveloppe supérieure)

Idée de l'algorithme

- Calculer la médiane
- Calculer le pont
- Le pont est un des segments de l'enveloppe supérieure
- Considérer deux sous-ensembles de points à partir desquels deux enveloppes supérieures sont calculées

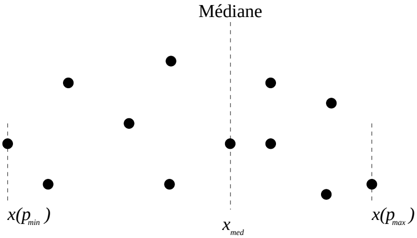
Stratégie couper-résoudre (enveloppe supérieure)



Idée de l'algorithme

- Calculer la médiane
- Calculer le pont
- Le pont est un des segments de l'enveloppe supérieure
- Considérer deux sous-ensembles de points à partir desquels deux enveloppes supérieures sont calculées

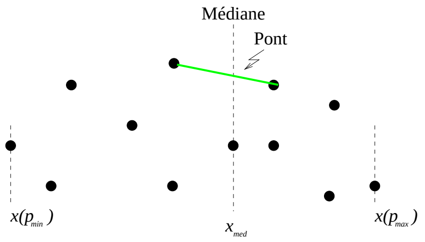
Stratégie couper-résoudre (enveloppe supérieure)



Idée de l'algorithme

- Calculer la médiane
- Calculer le pont
- Le pont est un des segments de l'enveloppe supérieure
- Considérer deux sous-ensembles de points à partir desquels deux enveloppes supérieures sont calculées

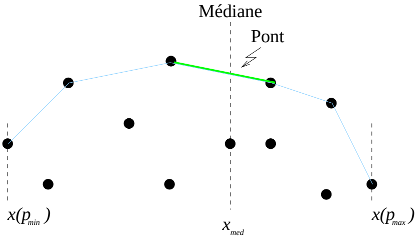
Stratégie couper-résoudre (enveloppe supérieure)



Idée de l'algorithme

- Calculer la médiane
- Calculer le pont
- Le pont est un des segments de l'enveloppe supérieure
- Considérer deux sous-ensembles de points à partir desquels deux enveloppes supérieures sont calculées

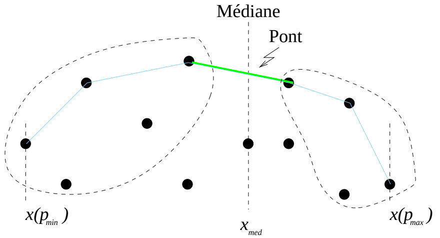
Stratégie couper-résoudre (enveloppe supérieure)



Idée de l'algorithme

- Calculer la médiane
- Calculer le pont
- Le pont est un des segments de l'enveloppe supérieure
- Considérer deux sous-ensembles de points à partir desquels deux enveloppes supérieures sont calculées

Stratégie couper-résoudre (enveloppe supérieure)



Idée de l'algorithme

- Calculer la médiane
- Calculer le pont
- Le pont est un des segments de l'enveloppe supérieure
- Considérer deux sous-ensembles de points à partir desquels deux enveloppes supérieures sont calculées

Stratégie couper-résoudre (exemple)

Donnée en entrée pour le calcul de l'enveloppe supérieure

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe supérieure

À partir d'un problème P de taille n :

Stratégie couper-résoudre (exemple)

Donnée en entrée pour le calcul de l'enveloppe supérieure

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe supérieure

À partir d'un problème P de taille n :

1. trouver le pont qui coupe la ligne verticale obtenue de la médiane des points (x_{med}) de S ;

Stratégie couper-résoudre (exemple)

Donnée en entrée pour le calcul de l'enveloppe supérieure

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe supérieure

À partir d'un problème P de taille n :

1. trouver le pont qui coupe la ligne verticale obtenue de la médiane des points (x_{med}) de S ;
2. considérer deux sous-ensembles de S, S_{left} les points à gauche de l'extrémité gauche du pont et S_{right} les points à droite de l'extrémité droite du pont ;

Stratégie couper-résoudre (exemple)

Donnée en entrée pour le calcul de l'enveloppe supérieure

L'ensemble de points $S = \{p_1, \ldots, p_n\}$.

Illustration pour le calcul de l'enveloppe supérieure

À partir d'un problème P de taille n :

1. trouver le pont qui coupe la ligne verticale obtenue de la médiane des points (x_{med}) de S ;
2. considérer deux sous-ensembles de S, S_{left} les points à gauche de l'extrémité gauche du pont et S_{right} les points à droite de l'extrémité droite du pont ;
3. calculer récursivement les enveloppes supérieures de S_{left} et S_{right}.

Stratégie couper-résoudre (exemple)

L'algorithme d'initialisation

1. `compute` min `and` max `such that for all` k, $1 \leq k \leq n$
   {Points extrêmes par rapport à l'abscisse}
2. $x(p_{min}) \leq x(p_k) \leq x(p_{max})$
3. $y(p_{min}) \geq y(p_k)$ `if` $x(p_{min}) = x(p_k)$
4. $y(p_{max}) \geq y(p_k)$ `if` $x(p_{max}) = x(p_k)$
5. `if` $min = max$ `then print(`min`) stop`
6. $T := \{p_{min}, p_{max}\} \cup \{p \in S \mid x(p_{min}) < x(p) < x(p_{max})\}$
7. $UpperHull(min, max, T)$

Remarque

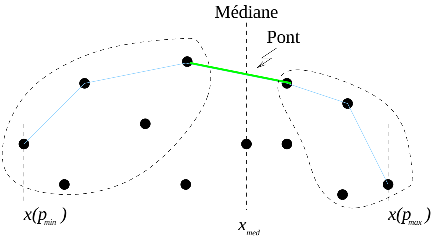
Les verticales aux extrémités sont omises.

Stratégie couper-résoudre (exemple)

Algorithme

1. function $UpperHull(l, r, S)$
2. compute x_{med} such that {Médiane par rapport à l'abscisse}
3. $x(p_i) \leq x_{med}$ for $\lceil |S|/2 \rceil$ points de S and
4. $x(p_i) \geq x_{med}$ for $\lfloor |S|/2 \rfloor$ points de S
   {Fixer comment couper et combiner}
5. $\langle i, j \rangle := Bridge(S, x_{med})$
   {Couper}
6. $S_{left} := \{p_i\} \cup \{p \in S \mid x(p) < x(p_i)\}$
7. $S_{right} := \{p_j\} \cup \{p \in S \mid x(p) > x(p_j)\}$
   {Résoudre}
8. if $i = l$ then print(i) else $UpperHull(l, i, S_{left})$
9. if $j = r$ then print(j) else $UpperHull(j, r, S_{right})$

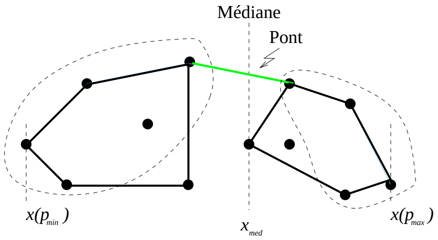
Stratégie couper-résoudre (première solution potentielle pour le calcul du pont)



Remarque

Si les enveloppes convexes de la partie de gauche et de la partie de droite étaient connues, la tangente commune aux deux parties pourrait être calculée en temps linéaire (c'est l'algorithme obtenu en utilisant la technique de diviser-pour-régner).

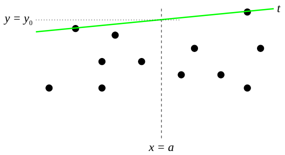
Stratégie couper-résoudre (première solution potentielle pour le calcul du pont)



Remarque

Si les enveloppes convexes de la partie de gauche et de la partie de droite étaient connues, la tangente commune aux deux parties pourrait être calculée en temps linéaire (c'est l'algorithme obtenu en utilisant la technique de diviser-pour-régner).

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

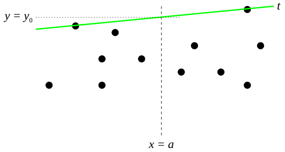


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

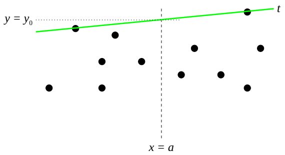


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

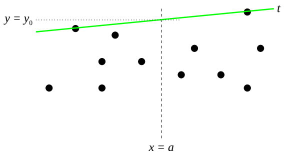


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

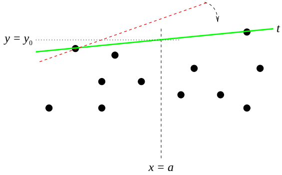


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

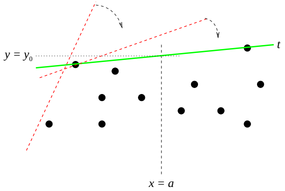


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

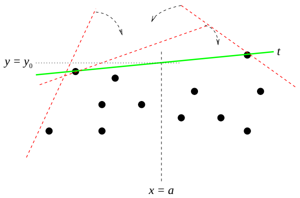


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)

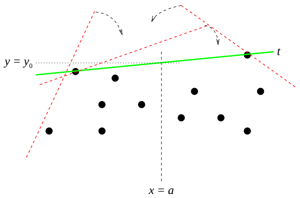


Formulation mathématique

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (deuxième solution potentielle pour le calcul du pont)



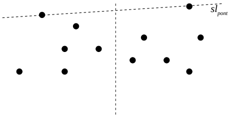
Formulation mathématique

$$\text{minimiser } \alpha a + \beta$$
$$\text{sujet à } \alpha x(p_i) + \beta \geq y(p_i)$$
$$\forall p_i \in S$$

Observations

- Considérons la droite t définie par $y = \alpha x + \beta$, qui coupe la médiane au point $x = a$.
- Considérons $y_0 = \alpha a + \beta$.
- Toute droite au-dessus de tous les points coupe la médiane à une ordonnée supérieure ou égale à y_0.

Stratégie couper-résoudre (calcul du pont)



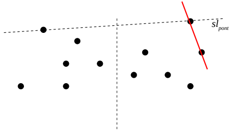
Lignes d'appui

Soit h une ligne d'appui de S avec une pente sl_h.

Résultat

- $sl_h < sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à droite de la médiane.
- $sl_h = sl_{pont}$ ssi h contient un point de S qui est strictement à droite de la médiane et un point de S qui est strictement à gauche de la médiane ou sur la médiane.
- $sl_h > sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à gauche de la médiane.

Stratégie couper-résoudre (calcul du pont)



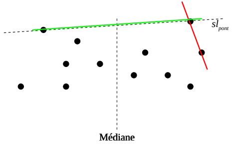
Lignes d'appui

Soit h une ligne d'appui de S avec une pente sl_h.

Résultat

- $sl_h < sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à droite de la médiane.
- $sl_h = sl_{pont}$ ssi h contient un point de S qui est strictement à droite de la médiane et un point de S qui est strictement à gauche de la médiane ou sur la médiane.
- $sl_h > sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à gauche de la médiane.

Stratégie couper-résoudre (calcul du pont)



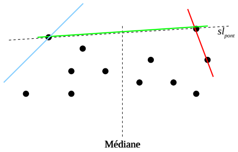
Lignes d'appui

Soit h une ligne d'appui de S avec une pente sl_h.

Résultat

- $sl_h < sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à droite de la médiane.
- $sl_h = sl_{pont}$ ssi h contient un point de S qui est strictement à droite de la médiane et un point de S qui est strictement à gauche de la médiane ou sur la médiane.
- $sl_h > sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à gauche de la médiane.

Stratégie couper-résoudre (calcul du pont)



Lignes d'appui

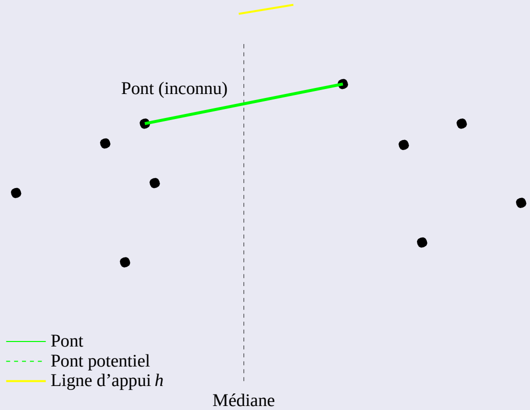
Soit h une ligne d'appui de S avec une pente sl_h.

Résultat

- $sl_h < sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à droite de la médiane.
- $sl_h = sl_{pont}$ ssi h contient un point de S qui est strictement à droite de la médiane et un point de S qui est strictement à gauche de la médiane ou sur la médiane.
- $sl_h > sl_{pont}$ ssi h contient seulement des points de S qui sont strictement à gauche de la médiane.

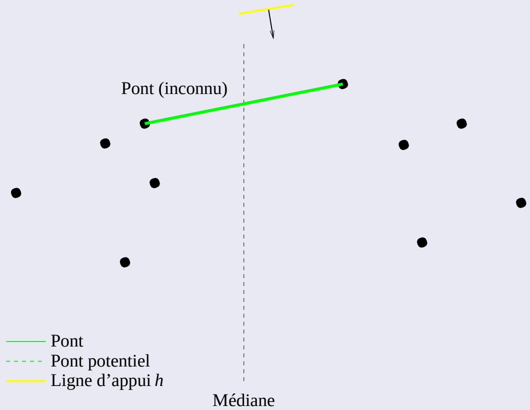
Stratégie couper-résoudre (élaguer des points)

h contient deux points de S de chaque côté de la médiane



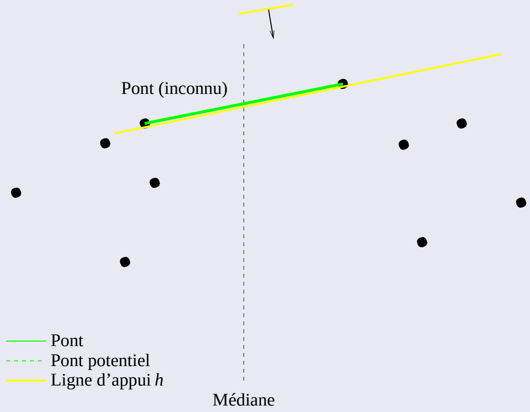
Stratégie couper-résoudre (élaguer des points)

h contient deux points de S de chaque côté de la médiane



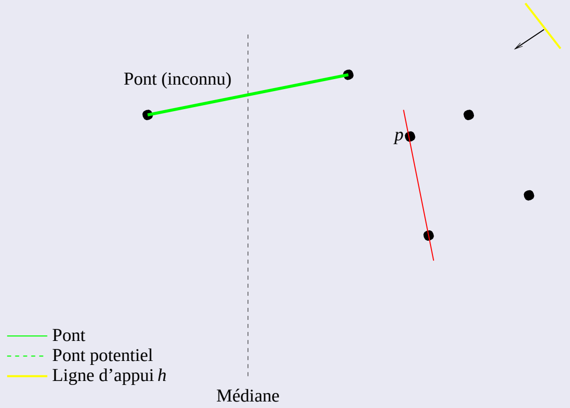
Stratégie couper-résoudre (élaguer des points)

h contient deux points de S de chaque côté de la médiane



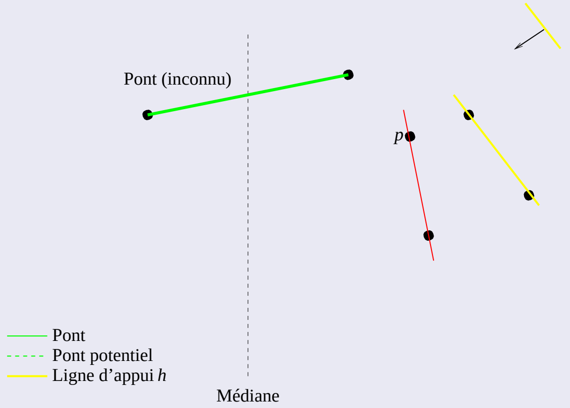
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à droite de la médiane



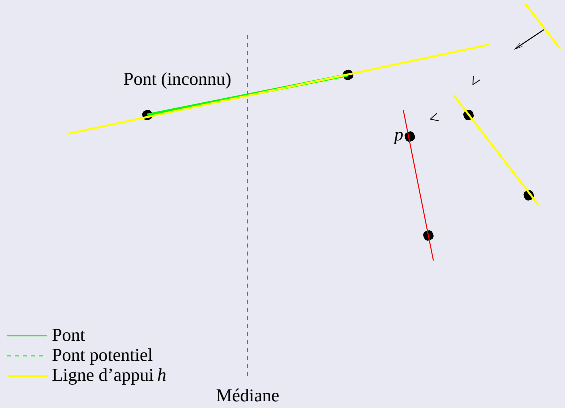
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à droite de la médiane



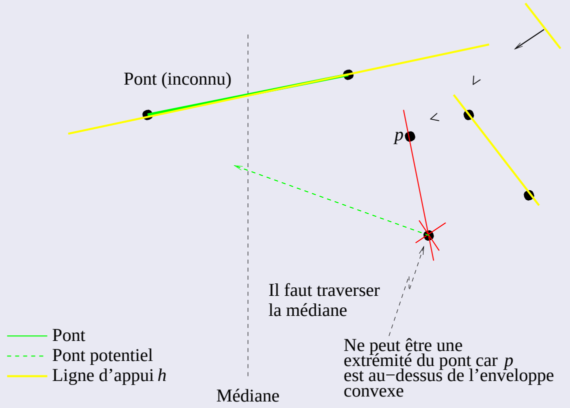
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à droite de la médiane



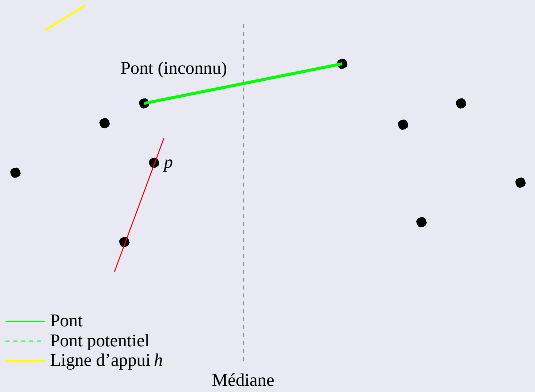
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à droite de la médiane



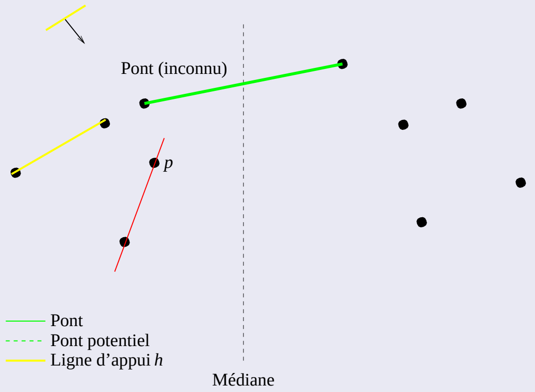
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à gauche de la médiane



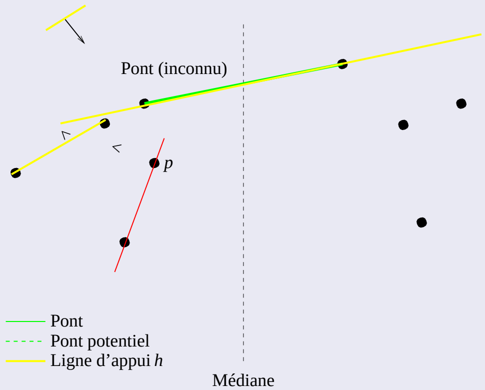
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à gauche de la médiane



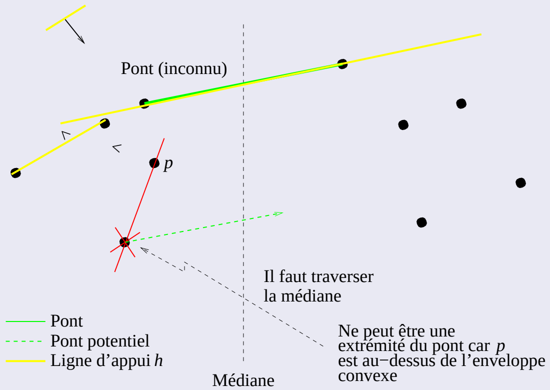
Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à gauche de la médiane

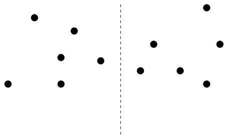


Stratégie couper-résoudre (élaguer des points)

h contient seulement des points de S à gauche de la médiane



Stratégie couper-résoudre (calcul du pont)

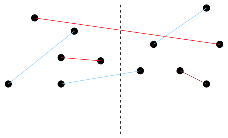


Élaguer des points

- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

Stratégie couper-résoudre (calcul du pont)

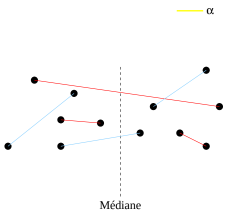


Élaguer des points

- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

Stratégie couper-résoudre (calcul du pont)

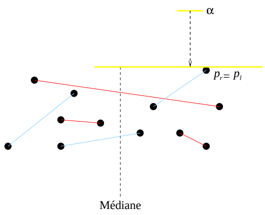


Élaguer des points

- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

Stratégie couper-résoudre (calcul du pont)

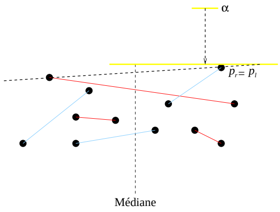


Élaguer des points

- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

Stratégie couper-résoudre (calcul du pont)

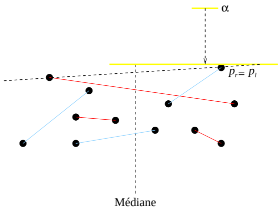


Élaguer des points

- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

Stratégie couper-résoudre (calcul du pont)

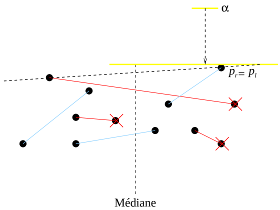


Élaguer des points

- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

Stratégie couper-résoudre (calcul du pont)

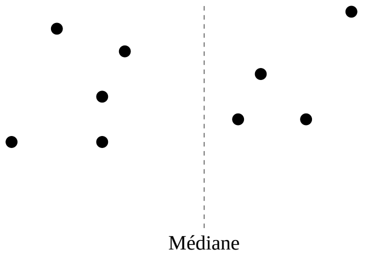


Élaguer des points

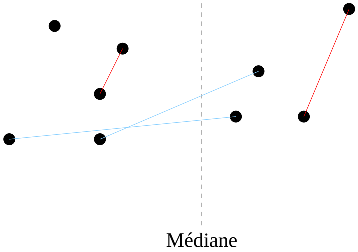
- Former des paires de points distinctes et calculer leur pente.
- Calculer la médiane des pentes, notée α.

- Déterminer la droite d'appui de pente α de S avec p_l et p_r comme extrémités.
- Comparer (virtuellement) la pente de la droite d'appui du pont avec α.
- Comparer toutes les pentes avec α.
- Élaguer des points.

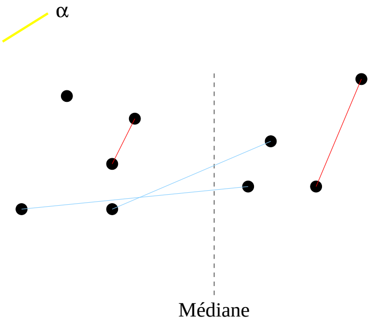
Stratégie couper-résoudre (calcul du pont)



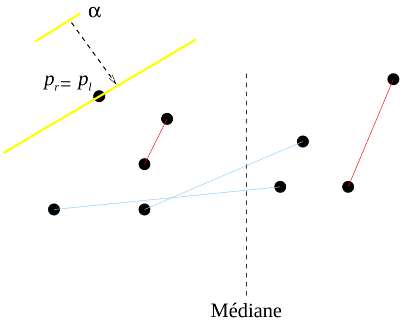
Stratégie couper-résoudre (calcul du pont)



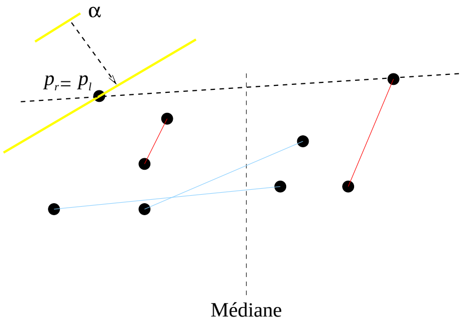
Stratégie couper-résoudre (calcul du pont)



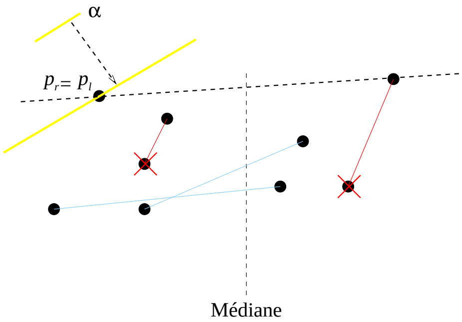
Stratégie couper-résoudre (calcul du pont)



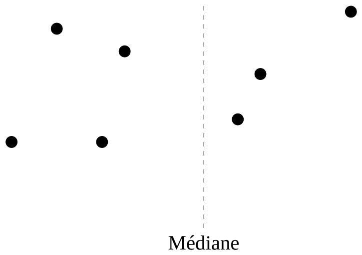
Stratégie couper-résoudre (calcul du pont)



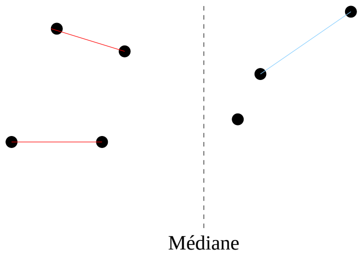
Stratégie couper-résoudre (calcul du pont)



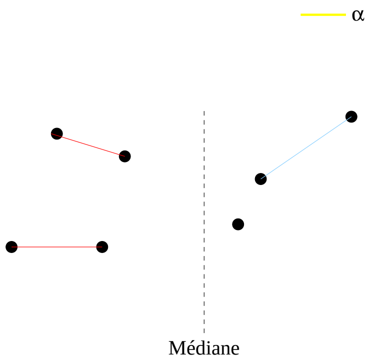
Stratégie couper-résoudre (calcul du pont)



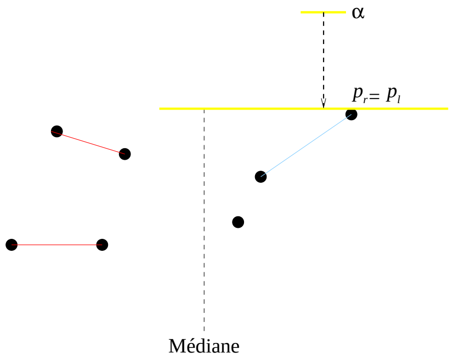
Stratégie couper-résoudre (calcul du pont)



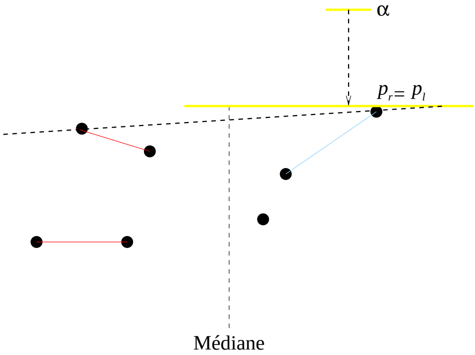
Stratégie couper-résoudre (calcul du pont)



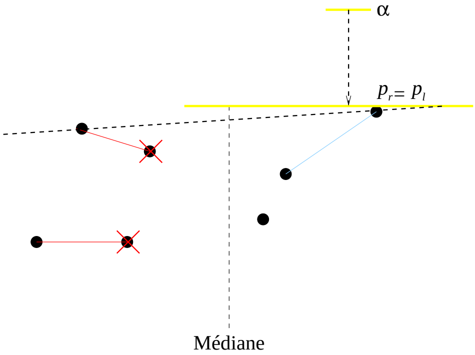
Stratégie couper-résoudre (calcul du pont)



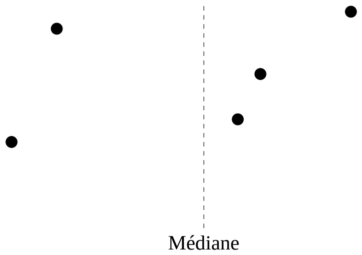
Stratégie couper-résoudre (calcul du pont)



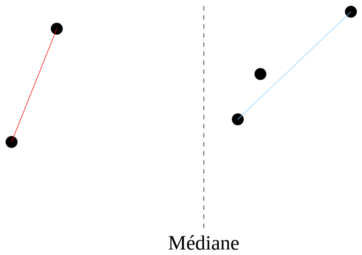
Stratégie couper-résoudre (calcul du pont)



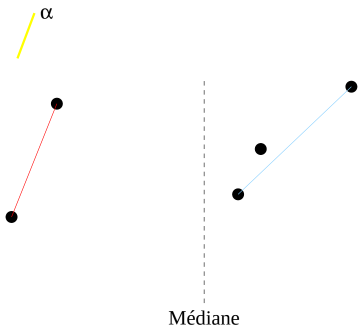
Stratégie couper-résoudre (calcul du pont)



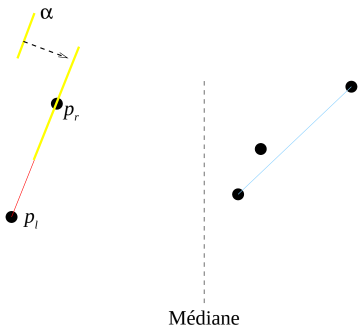
Stratégie couper-résoudre (calcul du pont)



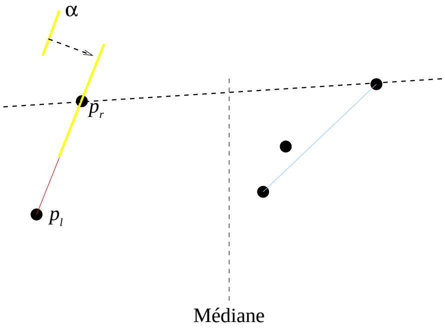
Stratégie couper-résoudre (calcul du pont)



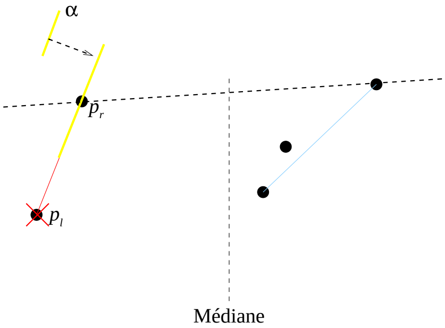
Stratégie couper-résoudre (calcul du pont)



Stratégie couper-résoudre (calcul du pont)



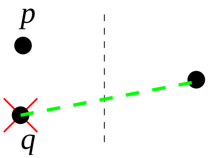
Stratégie couper-résoudre (calcul du pont)



Observation relative à l'enveloppe convexe supérieure

Éléments

Le segment en vert représente un pont potentiel.



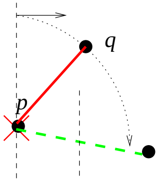
Résultat

Soit $\langle p, q \rangle$ une paire de points telle que $x(p) = x(q)$ et $y(p) > y(q)$. Alors q ne peut être une extrémité du pont (autrement p sera au-dessus de l'enveloppe convexe).

Observation relative à l'enveloppe convexe supérieure

Éléments

Le segment en vert représente un pont potentiel.



Résultat

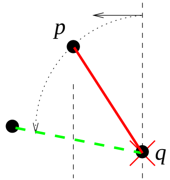
Soit $\langle p, q \rangle$ une paire de points telle que $x(p) < x(q)$ et soit $sl(p, q)$ la pente de la droite qui supporte le segment $\overline{pq}$. Alors,

- si $sl(p, q) > sl$, alors p ne peut être une extrémité du pont (autrement q sera au-dessus de l'enveloppe convexe).

Observation relative à l'enveloppe convexe supérieure

Éléments

Le segment en vert représente un pont potentiel.



Résultat

Soit $\langle p, q\rangle$ une paire de points telle que $x(p) < x(q)$ et soit $sl(p, q)$ la pente de la droite qui supporte le segment $\overline{pq}$. Alors,

- si $sl(p, q) < sl$, alors q ne peut être une extrémité du pont (autrement p sera au-dessus de l'enveloppe convexe).

Stratégie couper-résoudre (exemple)

Algorithme

function $Bridge(S, med)$

  if $|S| = 2$ then return $\langle i, j\rangle$ $\{S = \{p_i, p_j\} \wedge x(p_i) < x(p_j)\}$

  $\mathcal{S} := \{\langle p_i, p_j\rangle \in S^2 \mid x(p_i) \leq x(p_j)\} \cup \begin{cases} \emptyset & \text{si } |S| \mod 2 = 0 \\ \{p\} & \end{cases}$

   $\{$Pour tout $\langle p_i, p_j\rangle$ et $\langle p_k, p_l\rangle$ dans $\mathcal{S}$, $\{p_i, p_j\} \cap \{p_k, p_l\} = \emptyset\}$
   $\{p$ est l'unique point qui reste du choix des paires de points$\}$
   $\{$Points potentiels comme extrémités du pont$\}$

  $V := \begin{cases} \emptyset & \text{si } |S| \mod 2 = 0 \\ \{p\} & \end{cases}$

  for all $\langle p_i, p_j\rangle \in \mathcal{S}$ do

    if $x(p_i) = x(p_j)$ then

      $\mathcal{S} := \mathcal{S} - \{\langle p_i, p_j\rangle\}$ $\{$2 points à la verticale$\}$

      if $y(p_i) > y(p_j)$ then $V := V \cup \{p_i\}$

                    else $V := V \cup \{p_j\}$

Stratégie couper-résoudre (exemple)

Algorithme

$\texttt{function}$ $\mathit{Bridge}(S, \mathit{med})$

$\vdots$

  $\texttt{compute}$ α $\texttt{such that}$ {Médiane des pentes}

    $sl(p_i, p_j) \leq \alpha$ $\texttt{for}$ $\lceil |\mathcal{S}|/2 \rceil$ paires de points de $\mathcal{S}$ and

    $sl(p_i, p_j) \geq \alpha$ $\texttt{for}$ $\lfloor |\mathcal{S}|/2 \rfloor$ paires de points de $\mathcal{S}$

    {Partition des paires de points par rapport à la médiane}

  $\mathit{Small} := \{\langle p_i, p_j \rangle \in \mathcal{S} \mid sl(p_i, p_j) < \alpha\}$

  $\mathit{Equal} := \{\langle p_i, p_j \rangle \in \mathcal{S} \mid sl(p_i, p_j) = \alpha\}$

  $\mathit{Large} := \{\langle p_i, p_j \rangle \in \mathcal{S} \mid sl(p_i, p_j) > \alpha\}$

    {Trouver les points de la droite d'appui de pente α}

  $\mathit{Max} := \{p_i \in S \mid y(p_i) - (\alpha \times x(p_i))$ est maximum$\}$

  $p_l := \mathrm{argmin}(p \mid p \in \mathit{Max} : x(p))$

  $p_r := \mathrm{argmax}(p \mid p \in \mathit{Max} : x(p))$

Stratégie couper-résoudre (exemple)

Algorithme

```
function Bridge(S, med)
  ⋮
  if x(p_l) ≤ med and x(p_r) > med then
    {La droite d'appui contient le pont}
    return ⟨l, r⟩
  if x(p_r) ≤ med then   {Points tous à gauche de la médiane}
      {sl_pont < sl_droite_appui}
    for all ⟨p_i, p_j⟩ ∈ Large ∪ Equal do V := V ∪ {p_j}
    for all ⟨p_i, p_j⟩ ∈ Small do V := V ∪ {p_i, p_j}
  if x(p_l) > med then   {Points tous à droite de la médiane}
      {sl_pont > sl_droite_appui}
    for all ⟨p_i, p_j⟩ ∈ Small ∪ Equal do V := V ∪ {p_i}
    for all ⟨p_i, p_j⟩ ∈ Large do V := V ∪ {p_i, p_j}
  return Bridge(V, med)
```

Stratégie couper-résoudre (exemple)

Analyse de la complexité de la fonction *Bridge*

Au moins un quart des points de S sont éliminés à chaque appel de *Bridge* ; ces points ne sont pas dans V lors de l'appel récursif.

Équation de récurrence

$$T_{Bridge}(n) = \begin{cases} O(1) & \text{si } n = 2 \\ T_{Bridge}(\frac{3n}{4}) + O(n) & \text{si } n > 2 \end{cases}$$

À l'aide de la méthode d'itération

L'algorithme est en temps $O(n)$.

Explication

Stratégie couper-résoudre (exemple)

Analyse de la complexité de la fonction *UpperHull*

- Soit h le nombre de sommets de l'enveloppe supérieure.
- La partie d'initialisation se fait en temps linéaire ($O(n)$).

Équation de récurrence

$$T_{UpperHull}(n, h) \leq \begin{cases} cn & \text{si } h = 2 \\ cn + \max_{h_l + h_r = h} \left(T_{UpperHull}(\frac{n}{2}, h_l) + T_{UpperHull}(\frac{n}{2}, h_r)\right) & \text{si } h > 2 \end{cases}$$

où c est une constante positive et $n \geq h > 1$.

Solution devinée

$T_{UpperHull}(n, h)$ est en temps $O(n \lg h)$.

Stratégie couper-résoudre (exemple)

Analyse de la complexité de la fonction *UpperHull*

Prouver que $T_{UpperHull}(n,h) = cn \lg h$.

Méthode de substitution

Le cas de base $h = 2$ est satisfait, car $cn \lg 2 = cn$.

$$\begin{aligned}
T_{UpperHull}(n,h) &\leq cn + \max_{h_l+h_r=h} \left(c\frac{n}{2}\lg h_l + c\frac{n}{2}\lg h_r\right) \\
&= cn + \frac{1}{2}cn \max_{h_l+h_r=h} (\lg h_l h_r) \\
&= cn + \frac{1}{2}cn \lg(h/2)^2 \\
&= cn + cn \lg h/2 = cn + cn \lg h - cn = cn \lg h
\end{aligned}$$

Solution d'équations de récurrence à l'aide du théorème *maître*

Thérorème *maître*

Soit $a \geq 1$ et $b > 1$ des constantes réelles, soit $f(n) : \mathbb{N} \to \mathbb{R}^+$ et soit $T(n) : \mathbb{N} \to \mathbb{R}^+$ une fonction définie par l'équation de récurrence

$$T(n) = aT(\frac{n}{b}) + f(n),$$

où le terme $\frac{n}{b}$ est interprété comme $\lfloor \frac{n}{b} \rfloor$ ou $\lceil \frac{n}{b} \rceil$.

1. Si $f(n) \in O(n^{\log_b a - \epsilon})$ pour une constante $\epsilon > 0$, alors $T(n) \in \Theta(n^{\log_b a})$.
2. Si $f(n) \in \Theta(n^{\log_b a})$, alors $T(n) \in \Theta(n^{\log_b a} \lg n)$.
3. Si $f(n) \in \Omega(n^{\log_b a + \epsilon})$ pour une constante $\epsilon > 0$ et si $af(\frac{n}{b}) < cf(n)$ pour une constante $c < 1$ et tout n suffisamment grand, alors $T(n) \in \Theta(f(n))$.

Interprétation du théorème *maître*

- Les trois cas portent sur une comparaison des fonctions $f(n)$ et $n^{\log_b a}$ et celle qui est de *poids le plus fort* est retenue :
  - si la fonction $f(n)$ est asymptotiquement plus petite que $n^{\log_b a}$ par un facteur de n^{ϵ} (polynômialement plus petite) (cas 1), alors $T(n) \in \Theta(n^{\log_b a})$;
  - si la fonction $f(n)$ est asymptotiquement plus grande que $n^{\log_b a}$ par un facteur de n^{ϵ} (polynômialement plus grande) et qu'elle satisfait une condition de régularité (cas 3), alors $T(n) \in \Theta(f(n))$.
- Si les deux fonctions sont de même poids (cas 2), alors les deux fonctions sont multipliées par un facteur logarithmique et $T(n) \in \Theta(n^{\log_b a} \lg n) = \Theta(f(n) \lg n)$.

Interprétation du théorème *maître*

- Il y a un vide entre le cas 1 et le cas 2, c'est-à-dire que la fonction $f(n)$ peut être asymptotiquement plus petite que $n^{\log_b a}$, mais pas polynômialement plus petite.
- Il y a un vide entre le cas 2 et le cas 3, c'est-à-dire que la fonction $f(n)$ peut être asymptotiquement plus grande que $n^{\log_b a}$, mais pas polynômialement plus grande.

Solution d'équations de récurrence à l'aide d'une variante du théorème *maître*

Variante du thérorème *maître*

Soit $T(n) : \mathbb{N} \to \mathbb{R}^+$ une fonction croissante au sens large à partir d'un certain rang, telle qu'il existe des constantes entières $b \geq 2$ et $n_0 \geq 1$ et des constantes réelles $k \geq 0$, $a > 0$ et $d > 0$ pour lesquelles

$$T(n_0) = d \text{ et } T(n) = aT(\frac{n}{b}) + O(n^k)$$

pour tout $n > n_0$ avec n/n_0 une puissance de b.

1. Si $a > b^k$, alors $T(n) \in O(n^{\log_b a})$.
2. Si $a = b^k$, alors $T(n) \in O(n^k \log_b n)$.
3. Si $a < b^k$, alors $T(n) \in O(n^k)$.

Solution de l'équation de récurrence $CH(n) = 2CH(\frac{n}{2}) + O(n)$

Cas 2 de la variante du théorème *maître*

- $a = 2,\ b = 2,\ k = 1,\ n_0 = 2\,;$
- $CH(2) = 1.$

Remarque

$a = b^k$, car $2 = 2^1$.

Conclusion

$CH(n) \in O(n^1 \log_2 n) = O(n \lg n)$

Retour

Expansion de l'équation de récurrence

$$\begin{aligned} T(n) &= cn + T(\lfloor 3n/4 \rfloor) \\ &= cn + \lfloor 3cn/4 \rfloor + T(\lfloor 9n/16 \rfloor) = cn + \lfloor 3cn/4 \rfloor + \lfloor 9cn/16 \rfloor + T(\lfloor 27n/64 \rfloor) \\ &\qquad\qquad \text{remarque : } \lfloor \lfloor n/a \rfloor /b \rfloor = \lfloor \tfrac{n}{ab} \rfloor \\ &= cn + 3cn/4 + 9cn/16 + 27cn/64 + \cdots + cn\frac{3^{k-1}}{4^{k-1}} + T(\lfloor 3^k n/4^k \rfloor) \end{aligned}$$

Prenons le plus petit k tel que $\lceil \frac{n}{4^k} \rceil = 1$ (n est absorbé par 4^k), c'est-à-dire $4^k \approx n$. Ainsi $4^k \approx 4^{\log_4 n} = n$ (et $k \approx \log_4 n$).

Mais $3^k \approx 3^{\log_4 n} = n^{\log_4 3} = n^{0,7\cdots} \in o(n)$, car $a^{\log_b n} = n^{\log_b a}$ et $\lim_{n\to\infty} \frac{n^{0,7\cdots}}{n} = 0$ par la règle de l'Hospital. En effet $\lim_{n\to\infty} \frac{(0,7\cdots)n^{(0,7\cdots-1)}}{1} = \lim_{n\to\infty} \frac{0,7\cdots}{n^{0,2\cdots}} = 0$.

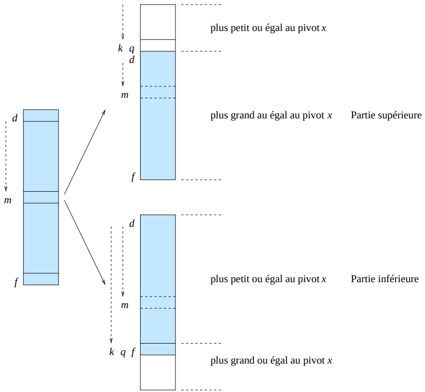
Ainsi, $T(\lfloor 3^k n/4^k \rfloor) \approx T(\lfloor (n^{0,7\cdots})\Theta(1) \rfloor)$.

$$\begin{aligned} T(n) &\approx cn + 3cn/4 + 9cn/16 + 27cn/64 + \cdots + cn\frac{3^{k-1}}{4^{k-1}} + c(n^{0,7\cdots})\Theta(1) + T(\cdot) \\ &\le cn\sum_{i=0}^{k-1}(3/4)^i + c(n^{0,7\cdots})\Theta(1)(1+\epsilon+\epsilon'+\epsilon''+\ldots) \\ &\le cn\sum_{k=0}^{\infty}(3/4)^k + o(n) \\ &= 4cn + o(n) \in O(n) \end{aligned}$$

Algorithme du calcul de la médiane

Éléments

- d, f et q sont des indices (absolus).
- m et k sont des déplacements par rapport à d.



Algorithme du calcul de la médiane

Algorithme

{Appel principal : $Randomized\_Select(A, 1, N, \lceil N/2 \rceil)$}
{d et f : indices pour le début et la fin du tableau A}
{m : déplacement par rapport à d qui, si A était trié, est la médiane}

```
1.  function Randomized_Select(A, d, f, m)
      {Cas de base (un seul élément) : la médiane est trouvée}
2.  if d = f then return A[d]
      {Division de A en parties, inférieure et supérieure, par rapport à q}
3.  q := Randomized_Partition(A, d, f)
      {Calcul du déplacement associé à q par rapport à d}
4.  k := q − d + 1   {q + 1 = k + d : si m est relatif à d,}
                     {                alors m − k est relatif à q + 1}
5.  if m ≤ k then
        {La médiane est dans la partie inférieure}
6.      return Randomized_Select(A, d, q, m)
7.  else
        {La médiane est dans la partie supérieure}
8.      return Randomized_Select(A, q + 1, f, m − k)
```

Algorithme du calcul de la médiane

Choix du pivot, échange et partition

1. **function** *Randomized_Partition*(A, d, f)
   {Le pivot est choisi aléatoirement}
2. $i := Random(d, f)$
   {Le pivot est échangé avec $A[d]$, donc le pivot est dans $A[d]$}
3. $Swap(A, d, i)$
   {Partition de A par rapport au pivot}
4. **return** $Partition(A, d, f)$

Algorithme du calcul de la médiane

Échanges successifs d'éléments

1. `function` $Partition(A, d, f)$
   $\{x$ est le pivot$\}$
   $\{i$ est au-dessus du début et j est au-dessous de la fin du tableau$\}$
2. $x := A[d] \quad i := d - 1 \quad j := f + 1$
3. `while true`
   {Recherche d'un élément plus petit ou égal au pivot depuis le bas}
4. `repeat` $j := j - 1$ `until` $A[j] \leq x$
   {Recherche d'un élément plus grand ou égal au pivot depuis le haut}
5. `repeat` $i := i + 1$ `until` $A[i] \geq x$
   {Échange d'un plus petit élément avec un plus grand ou retour}
6. `if` $i < j$ `then` $Swap(A, i, j)$ `else return` j

Algorithme du calcul de la médiane

Éléments

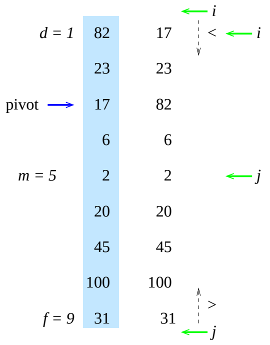
$d = 1$, $f = 9$, $m = 5$ et $i = 3$ (le pivot est 17 par hasard)

$d = 1$	82
	23
pivot $\longrightarrow$	17
	6
$m = 5$	2
	20
	45
	100
$f = 9$	31

Algorithme du calcul de la médiane

Éléments

$d = 1$, $f = 9$, $m = 5$ et $i = 3$ (le pivot est 17 par hasard)



Algorithme du calcul de la médiane

Éléments

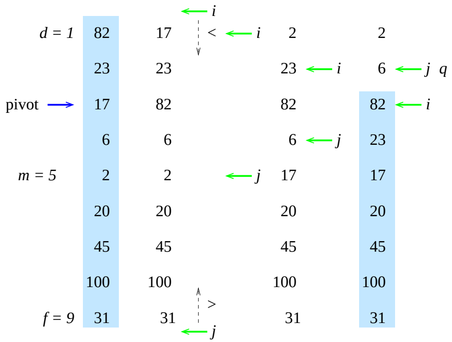
$d = 1$, $f = 9$, $m = 5$ et $i = 3$ (le pivot est 17 par hasard)

			← i			
$d = 1$	82	17	<	← i	2	
	23	23			23	← i
pivot →	17	82			82	
	6	6			6	← j
$m = 5$	2	2		← j	17	
	20	20			20	
	45	45			45	
	100	100			100	
$f = 9$	31	31	>		31	
			← j			

Algorithme du calcul de la médiane

Éléments

$d = 1$, $f = 9$, $m = 5$ et $i = 3$ (le pivot est 17 par hasard)



Algorithme du calcul de la médiane

Éléments

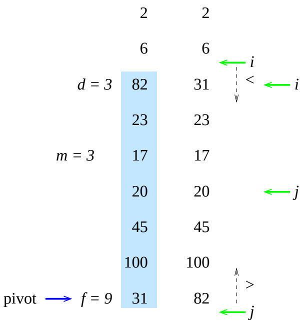
$d = 3$, $f = 9$, $m = 3$ et $i = 9$ (le pivot est 31 par hasard)

		2
		6
	$d = 3$	82
		23
	$m = 3$	17
		20
		45
		100
pivot $\longrightarrow$	$f = 9$	31

Algorithme du calcul de la médiane

Éléments

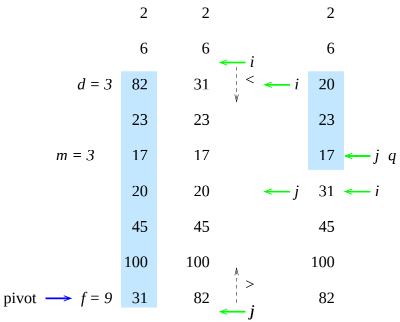
$d = 3$, $f = 9$, $m = 3$ et $i = 9$ (le pivot est 31 par hasard)



Algorithme du calcul de la médiane

Éléments

$d = 3$, $f = 9$, $m = 3$ et $i = 9$ (le pivot est 31 par hasard)



Algorithme du calcul de la médiane

Éléments

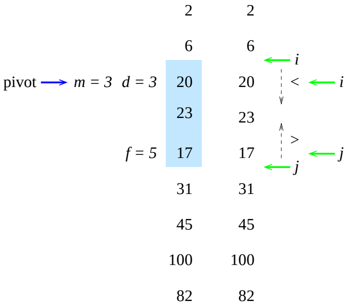
$d = 3$, $f = 5$, $m = 3$ et $i = 3$ (le pivot est 20 par hasard)

	2
	6
pivot $\longrightarrow$ $m = 3$ $d = 3$	20
	23
$f = 5$	17
	31
	45
	100
	82

Algorithme du calcul de la médiane

Éléments

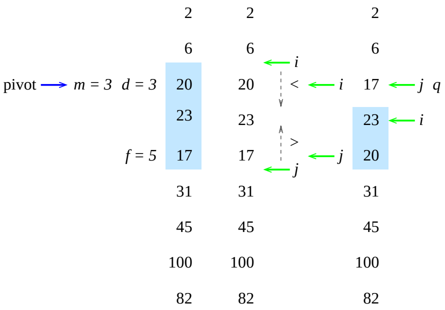
$d = 3$, $f = 5$, $m = 3$ et $i = 3$ (le pivot est 20 par hasard)



Algorithme du calcul de la médiane

Éléments

$d = 3$, $f = 5$, $m = 3$ et $i = 3$ (le pivot est 20 par hasard)



Algorithme du calcul de la médiane

Éléments

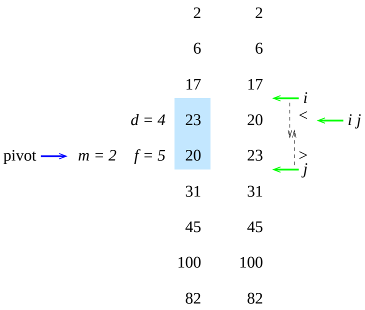
$d = 4$, $f = 5$, $m = 2$ et $i = 5$ (le pivot est 20 par hasard)

			2
			6
			17
		$d = 4$	23
pivot $\longrightarrow$	$m = 2$	$f = 5$	20
			31
			45
			100
			82

Algorithme du calcul de la médiane

Éléments

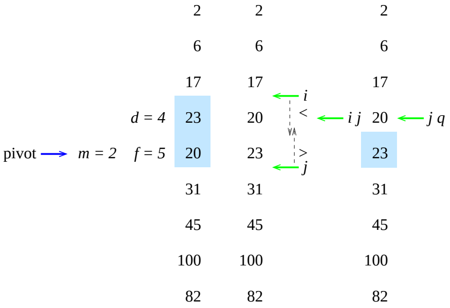
$d = 4$, $f = 5$, $m = 2$ et $i = 5$ (le pivot est 20 par hasard)



Algorithme du calcul de la médiane

Éléments

$d = 4$, $f = 5$, $m = 2$ et $i = 5$ (le pivot est 20 par hasard)



Séries géométriques

Série géométrique ou exponentielle, cas fini

$1 + x + x^2 + \cdots + x^n = \sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$ si $x \in \mathbb{R}$ et $x \neq 1$

Série géométrique ou exponentielle, cas infini

$\sum_{k=0}^{\infty} x^k = \frac{1}{1-x}$ si $|x| < 1$ (série décroissante)

Dérivée de la série précédente et multiplication par x

$\sum_{k=0}^{\infty} kx^k = \frac{x}{(1-x)^2}$ si $|x| < 1$

Série arithmétique et géométrique

$\sum_{k=0}^{n-1}(a + kd)r^k = \frac{a(1-r^n)}{1-r} + \frac{rd(1-nr^{n-1}+(n-1)r^n)}{(1-r)^2}$

Retour

Logarithme

Notations

- $\lg n = \log_2 n$ (logarithme binaire)
- $\ln n = \log_e n$ (logarithme népérien)
- $\lg^k n = (\lg n)^k$ (puissance de)
- $\lg \lg n = \lg(\lg n)$ (composition)
- $\lg n + k = (\lg n) + k$

Logarithme

Propriétés

Pour $a > 0$, $b > 0$, $c > 0$ et $a, b, c, n \in \mathbb{R}$

- $a = b^{\log_b a}$
- $\log_c(ab) = \log_c a + \log_c b$
- $\log_b(1/a) = -\log_b a$
- $\log_b a^n = n \log_b a$
- $\log_b a = \frac{\log_c a}{log_c b}$
- $\log_b a = \frac{1}{\log_a b}$
- $a^{\log_b n} = n^{\log_b a}$

Troisième partie III

Bibliographie

Lectures auxiliaires

- T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein.
  Introduction to Algorithms, Second Edition.
  The MIT Press, chapitre 9 (pour le calcul de la médiane), chapitre 33, 183-196, 933–965, 2001.
- D. G. Kirkpatrick, R. Seidel.
  The ultimate planar convex hull algorithm ?.
  SIAM Journal on Computing, 15 (1), 1986, 287–299.
- F. P. Preparata, S. J. Hong.
  Convex hulls of finite sets of points in two and three dimensions.
  Communications of the ACM, 20 (2), 1977, 87–93.

Quatrième partie IV

Feuille de route

Travail personnel

Lectures

- lecture des transparents
- lecture du chapitre 33 du livre du cours
- lectures auxiliaires pour en savoir plus

Exercices

Liste d'exercices #6 du répertoire public du cours

Devoir

Devoir #5