SOMMAIRE

- Informations pratiques
- Introduction
- Notions Algorithmiques
- Méthodes Algorithmiques pour la géométrie
- Modéliser *le monde*
- Méthodes géométriques
  - Notions de base en géométrie
  - Méthodes applicables aux modèles discrets
  - Méthodes applicables aux modèles continus

Méthodes algorithmiques pour la géométrie

- Algos géométriques font largement appel aux techniques classiques de l'Algorithmique
  - 3 méthodes classiques
    - Méthode incrémentale
    - Division/Fusion
    - Le Balayage

Méthodes algorithmiques pour la géométrie

- Méthode incrémentale
  - Consiste à traiter 1 par 1 les données
  - Algo est initialisé en construisant la solution du pb correspondant à un sous-ensemble de l'ensemble de données
  - Conserve cette solution lorsque les autres données sont introduites successivement
  - Données peuvent être triées au départ pour tirer profit de cet ordre

Méthodes algorithmiques pour la géométrie

- Méthode incrémentale
  - Mais l'ordre de traitement des données peut être
    - indifférent
    - ou volontairement aléatoire ⇔ version randomisée de la méthode incrémentale
      - Déroulement comporte une part de hasard (choix aléatoires faits à certains moments qui conditionnent le déroulement ultérieur)
      - Algorithme fournit toujours une solution exacte
      - mais le nb d'opérations nécessaires à son déroulement complet dépend des choix effectués
      - Performances s'évaluent en moyenne sur tous les choix possibles au niveau du déroulement de l'algo=> analyse randomisée
      - Mais pas d'hypothèses sur l'ensemble des données (on prend le cas le pire)

Méthodes algorithmiques pour la géométrie

- Division et fusion
  - Une des plus vieilles méthodes algorithmiques dont le champ d'application dépasse celui de la géométrie
  - Application du vieil adage politique "Diviser pour Régner"
    D'où son nom anglophone " Divide and Conquer "
  - Application récursive du schéma algorithmique suivant
    - Division du pb en sous-pbs de tailles inférieures
    - Résolution séparée de chaque sous-pb par application récursive de la même méthode
    - Fusion des solutions des sous-pbs pour construire la solution du pb initial

Méthodes algorithmiques pour la géométrie

- Division et fusion (suite)
  - Analyse de l'efficacité dépend de
    - La complexité des étapes de division et de fusion
    - La taille et du nombre de sous-problèmes engendrés à chaque étape

      ☞ si pb de taille n est divisé en p sous-pbs de taille n/q (n et q : cstes entières)

      et

      si étapes de division et fusion nécessitent f(n) opération élémentaires

      Alors la complexité de l'algorithme de division et fusion est donnée par l'équation de récurrence

$$t(n)=p\,t\left(\frac{n}{q}\right)+f(n)$$ Au rang k on aura $$t(n)=p^k\,t\left(\frac{n}{q^k}\right)+\sum_{j=0}^{k-1}p^j\,f\left(\frac{n}{q^j}\right)$$

Méthodes algorithmiques pour la géométrie

- Division et fusion /Analyse de l'efficacité (suite)
  - En général le processus de division récursive s'arrête quand le sous-problème devient trivial (taille < n_0)
  - Si $k = \lfloor \log_q \left(\frac{n}{n_0}\right) \rfloor$

    Alors $t(n) = \theta\left(p^k + \sum_{j=0}^{k-1} p^j f\left(\frac{n}{q^j}\right)\right)$

    Résolution de tous les sous problèmes triviaux

    Complexité totale de toutes les étapes de division et fusion exécutées par l'algo

Méthodes algorithmiques pour la géométrie

- Division et fusion /Analyse de l'efficacité (suite)

$$t(n) = p^k \; t\left(\frac{n}{q^k}\right) + \sum_{j=0}^{k-1} p^j \; f\left(\frac{n}{q^j}\right)$$

  - Si f est une fonction multiplicative (f(xy)=f(x)*f(y))

    Alors

$$t(n) = \theta\left(p^k \; t\left(\frac{n}{q^k}\right) + f(n) \sum_{j=0}^{k-1} \frac{p^j}{f(q^j)} \right)$$

Méthodes algorithmiques pour la géométrie

$$t(n) = \theta\left(p^k \; t\left(\frac{n}{q^k}\right) + f(n)\sum_{j=0}^{k-1}\frac{p^j}{f(q^j)} \right)$$

- Si on remarque que $n = q^{(\log n / \log q)}$ on a

$$t(n) = \theta\left(n^{\frac{\log p}{\log q}} + n^{\frac{\log f(q)}{\log q}} \sum_{j=0}^{k-1}\frac{p^j}{f(q^j)} \right)$$

- D'où
  - $t(n) = \Theta(n^{(\log p / \log q)})$ si $p > f(q)$
  - $t(n) = \Theta(n^{(\log p / \log q)} \log n)$ si $p = f(q)$
    - Si $f(n) = n^\alpha$ alors $t(n) = \Theta(n^\alpha \log n)$
  - $t(n) = \Theta(n^{(\log f(q) / \log q)} \log n)$ si $p < f(q)$
    - Si $f(n) = n^\alpha$ alors $t(n) = \Theta(n^\alpha)$

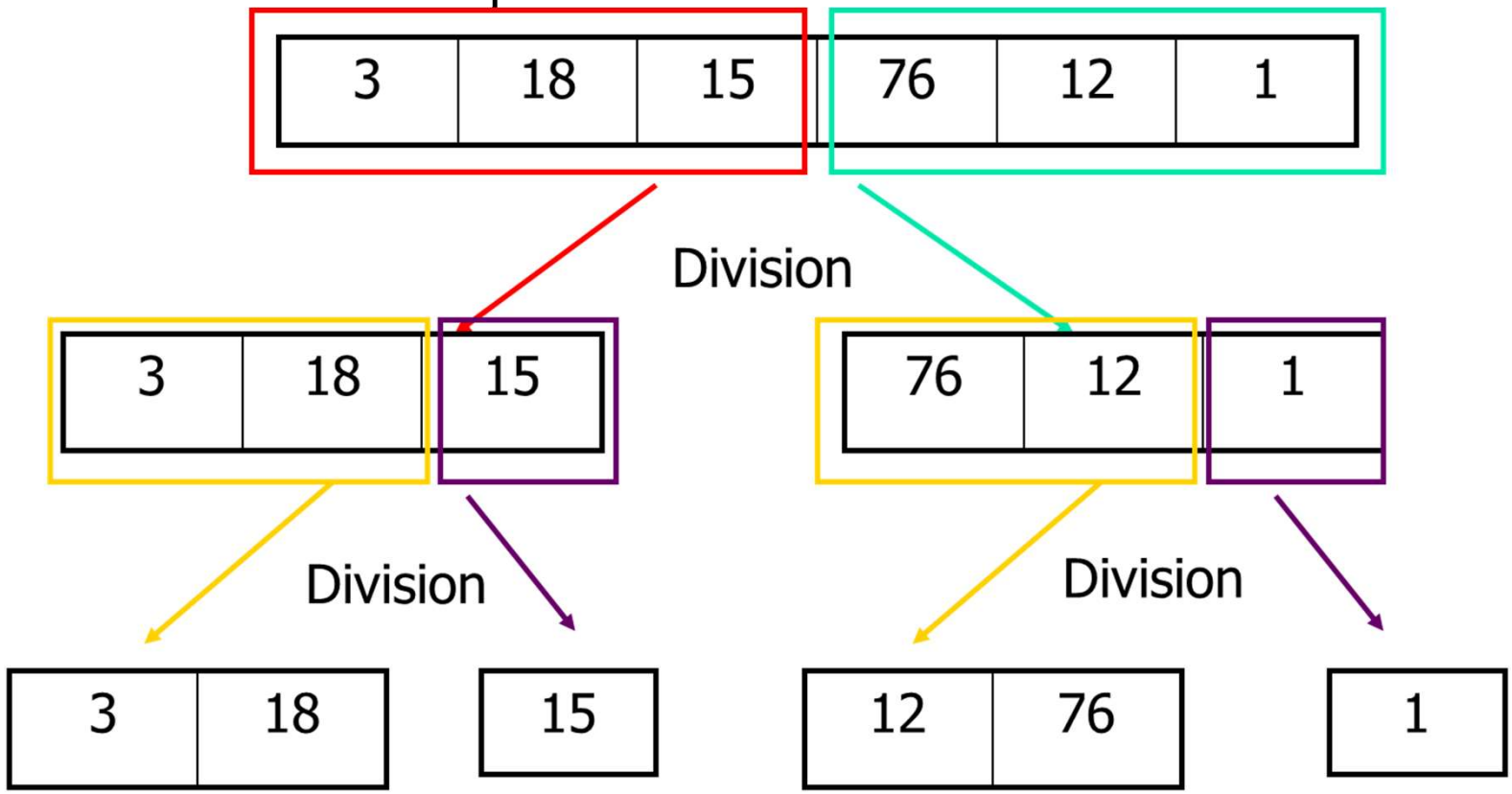
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple
  - Algo de Tri – fusion
    - Soit une suite finie $X = (x_1, x_2, \ldots, x_n)$
    - Pb consiste à construire la suite Y formée par les éléments de X classés par ordre croissant
      - Division : la liste X est divisée en 2 sous listes X_1 et X_2 de taille n/2
      - Résolution: chacune des sous suites est triée en appliquant récursivement la même méthode Y_1 et Y_2 sont les 2 suites résultant du tri des sous suites X_1 et X_2 .
      - Fusion : On obtient la suite Y en fusionnant les 2 sous-suites Y_1 et Y_2

        On parcourt simultanément les 2 sous-suites et à chaque pas on compare les éléments courants des 2 suites, et le plus petit des 2 est ajouté à la liste Y

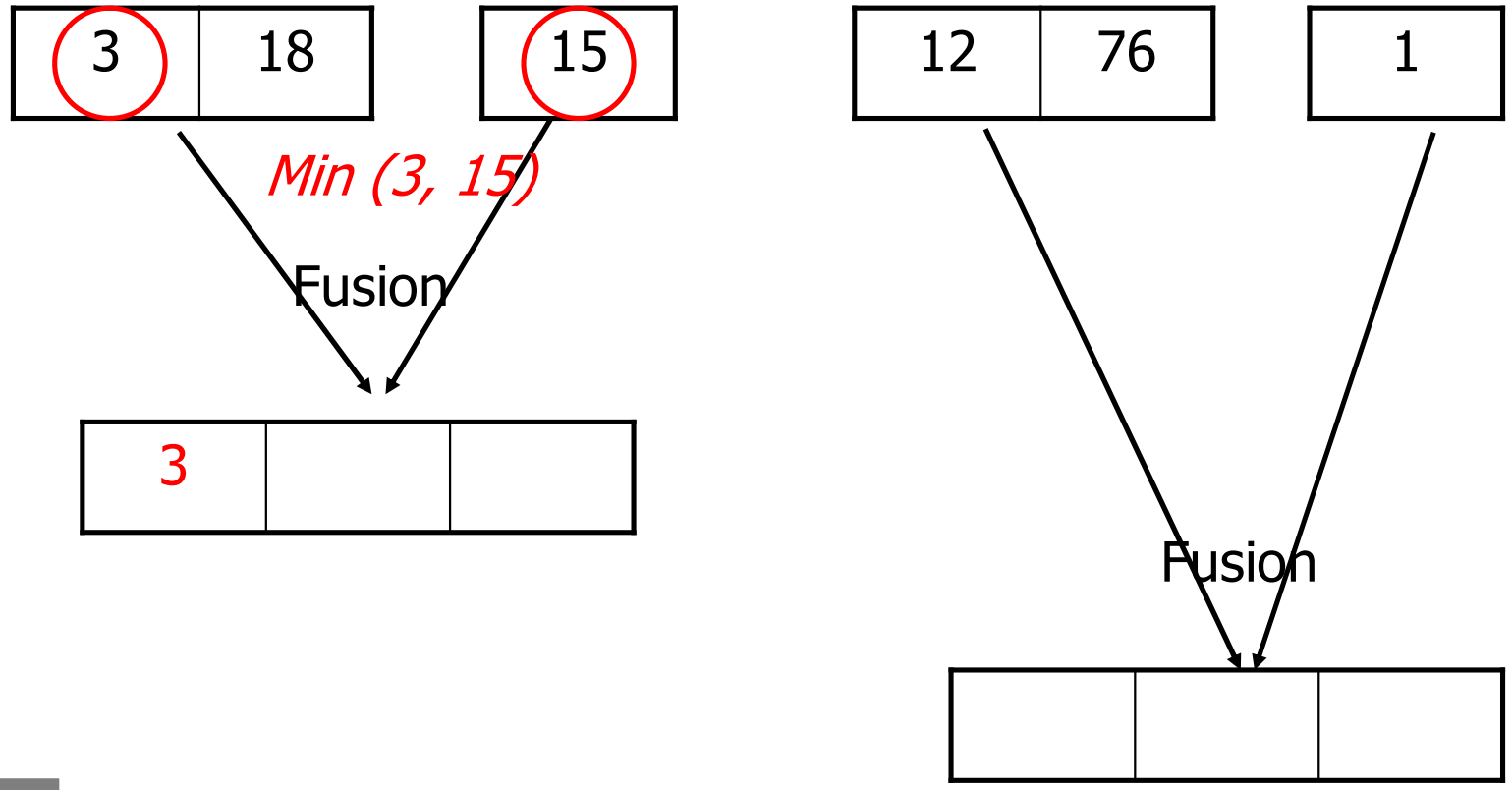
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



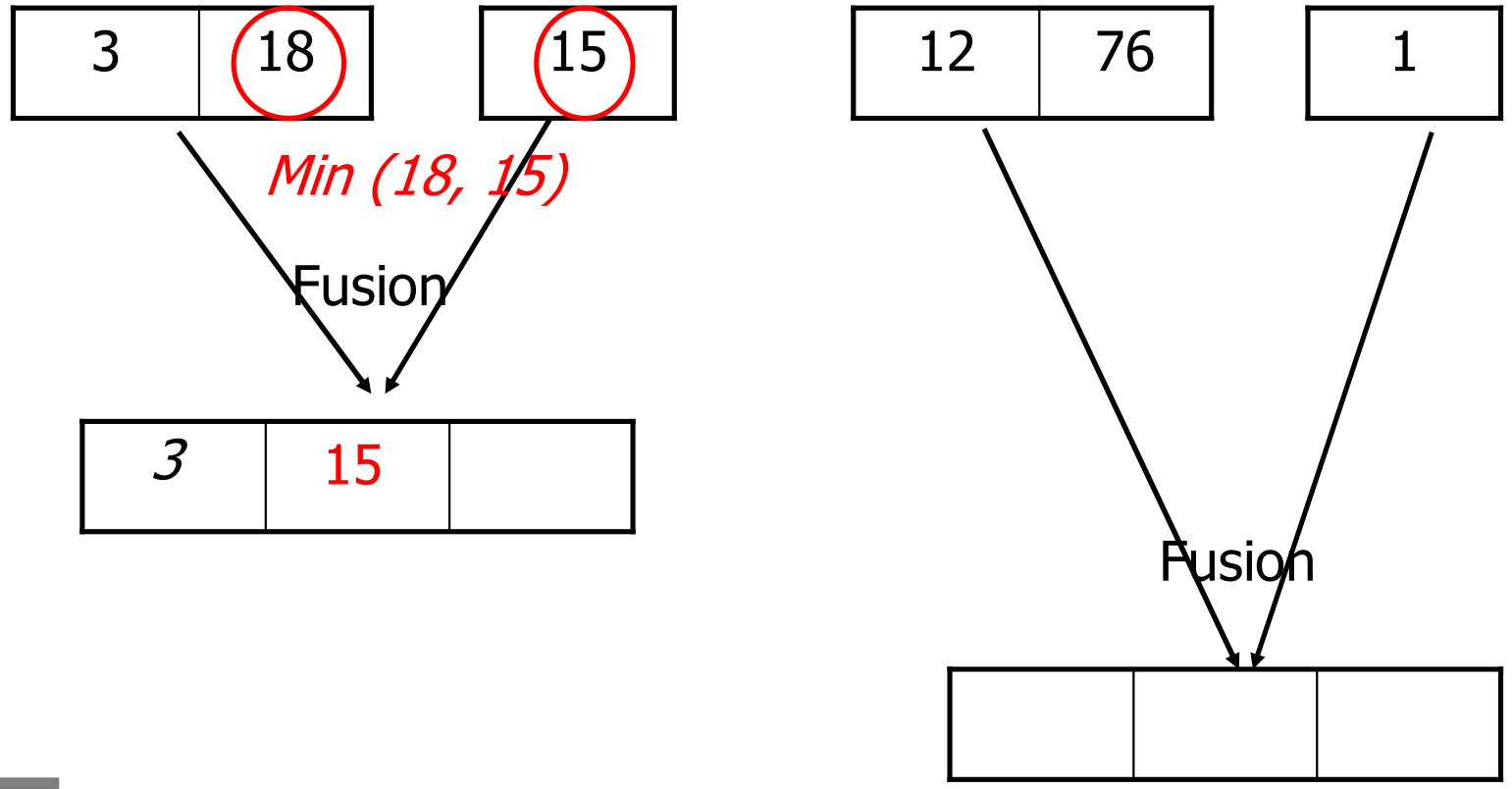
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



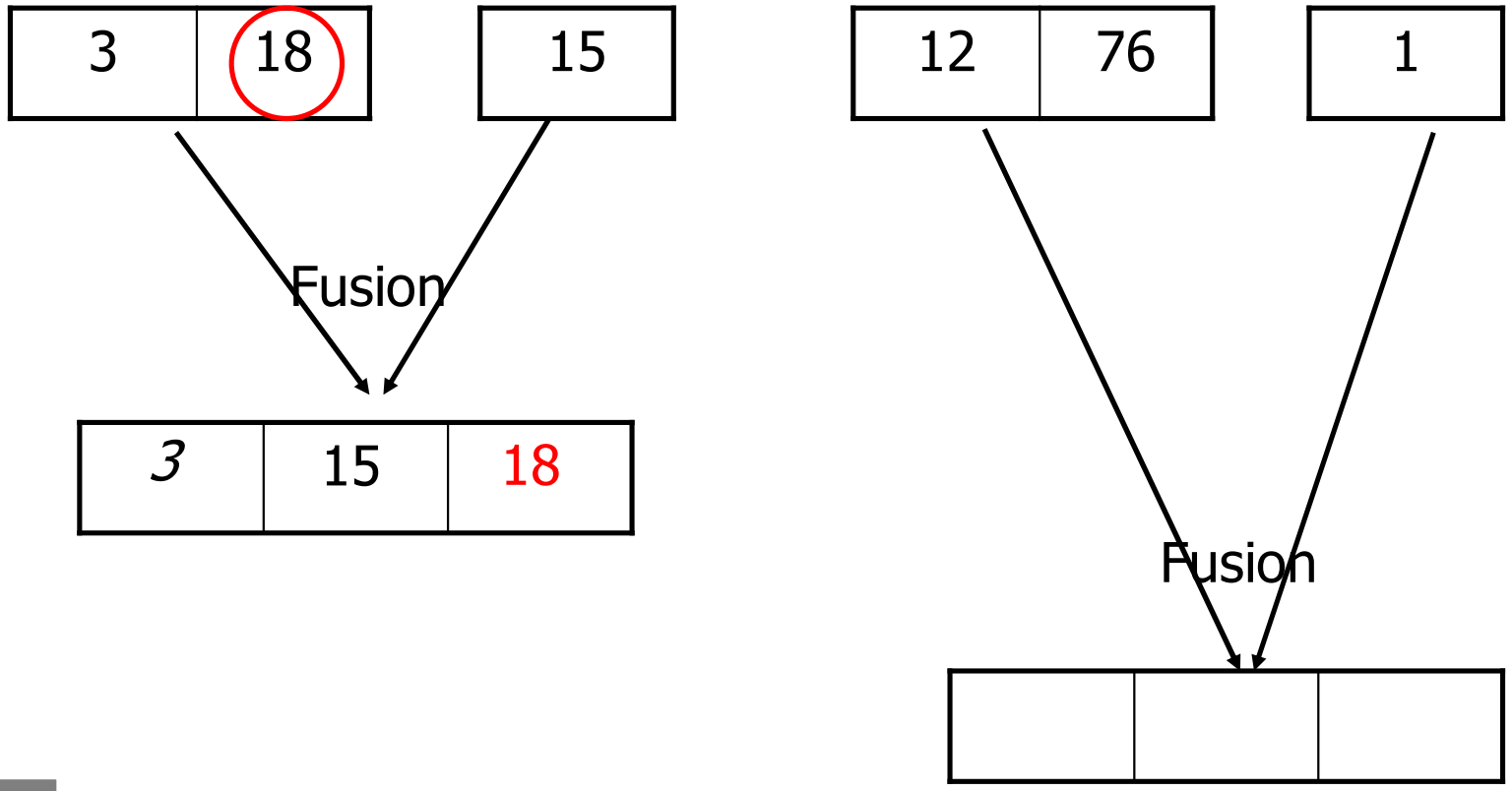
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



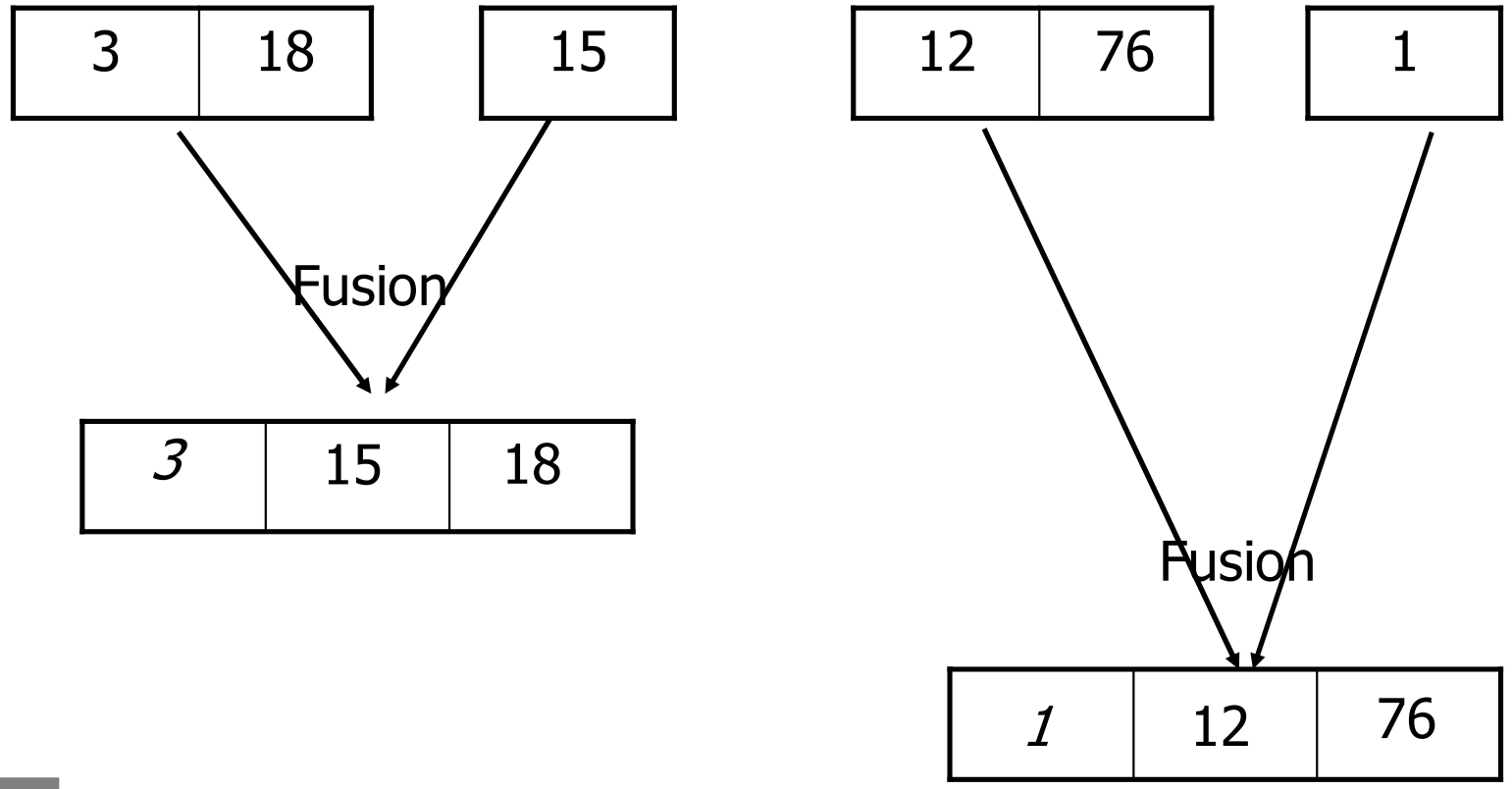
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



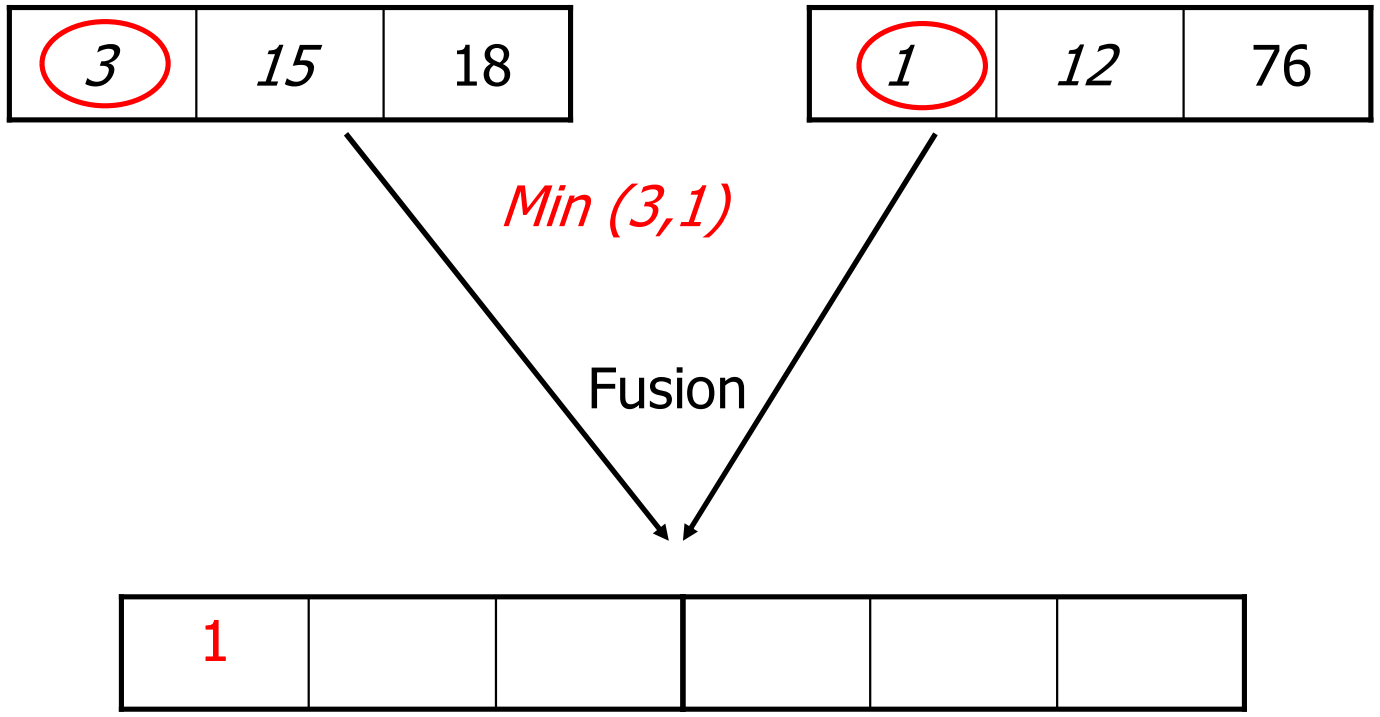
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



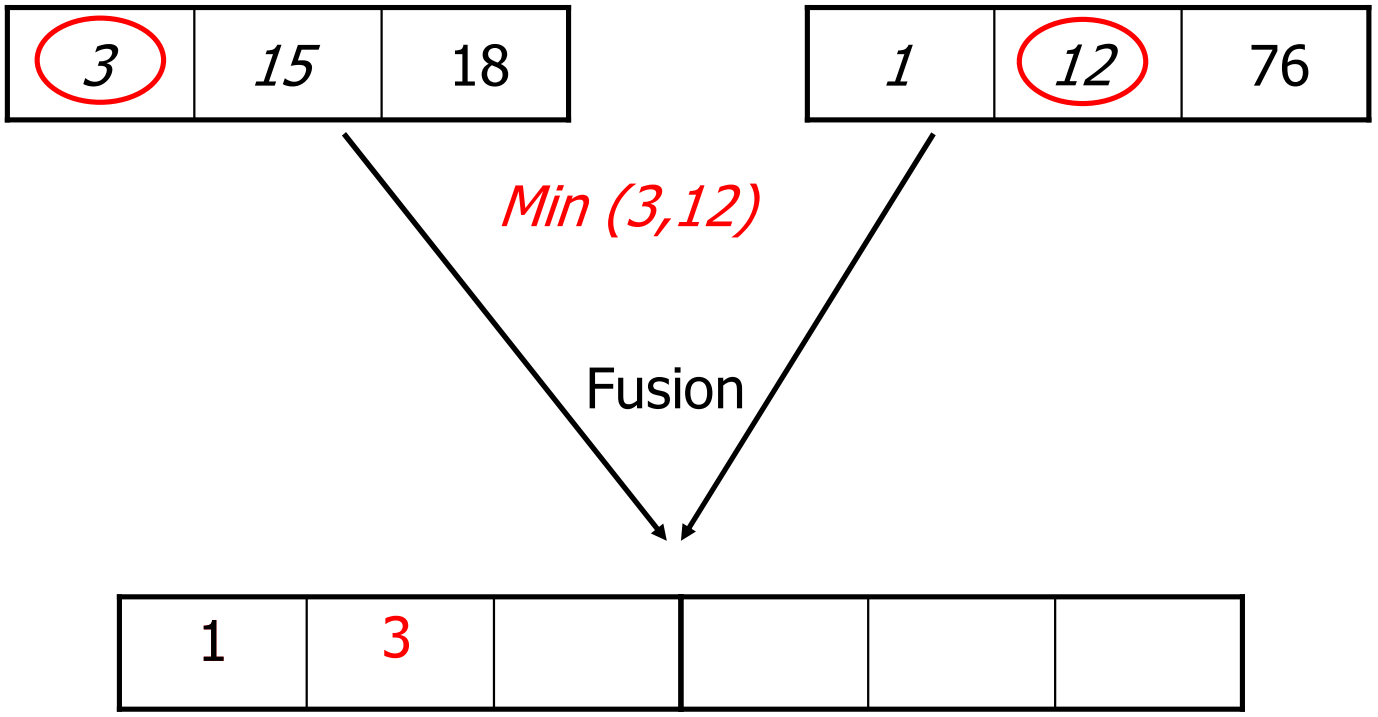
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



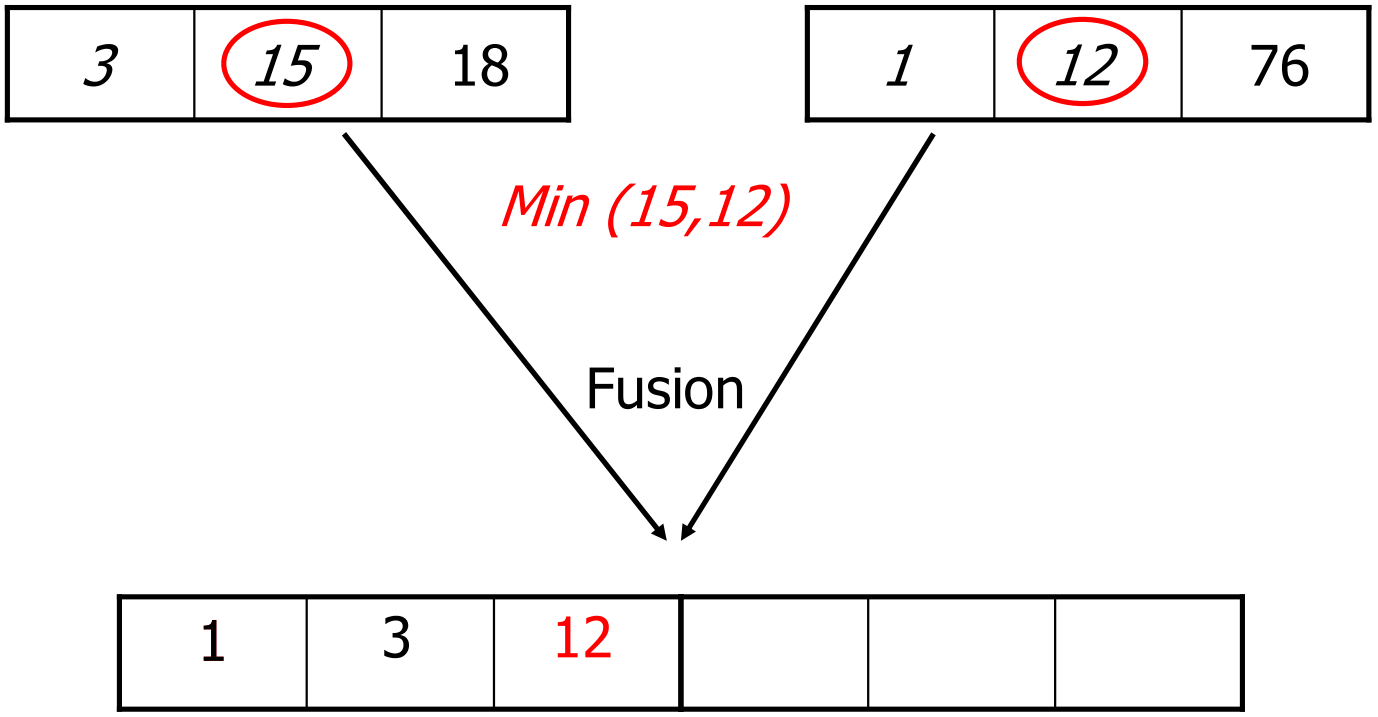
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



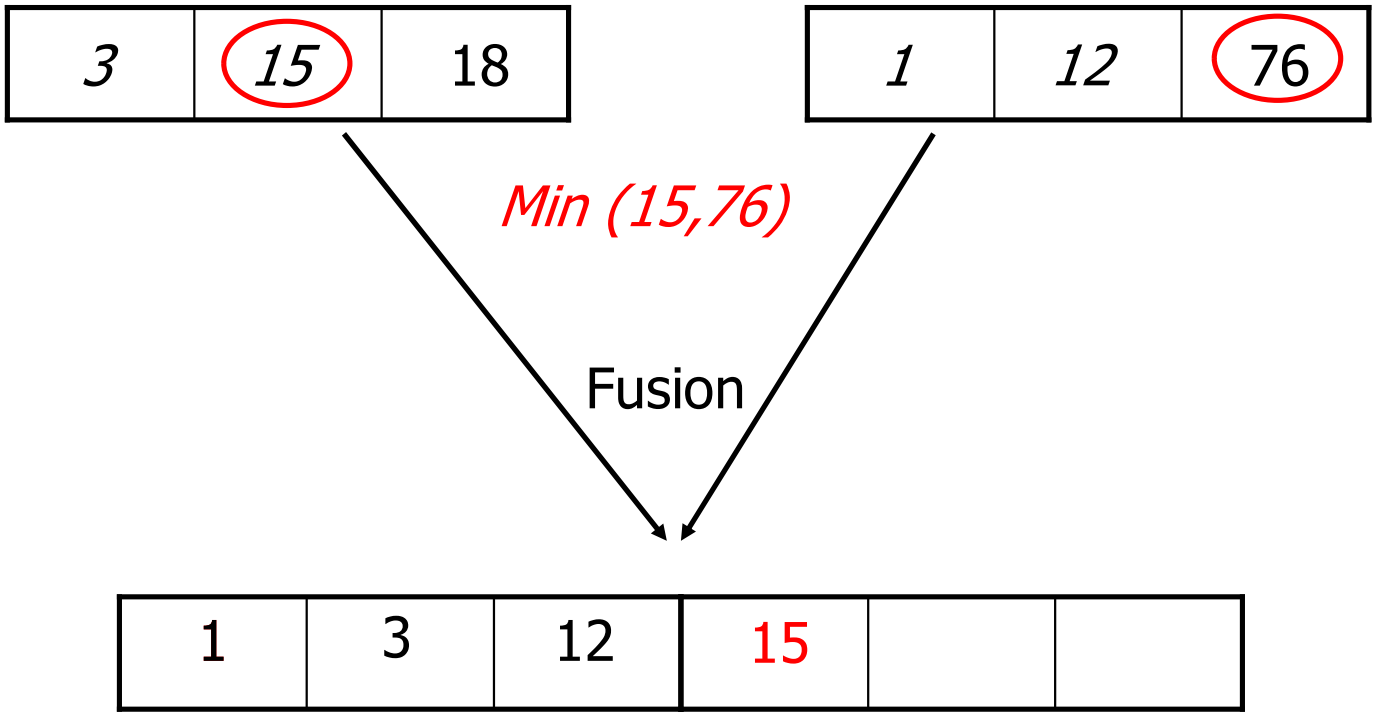
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



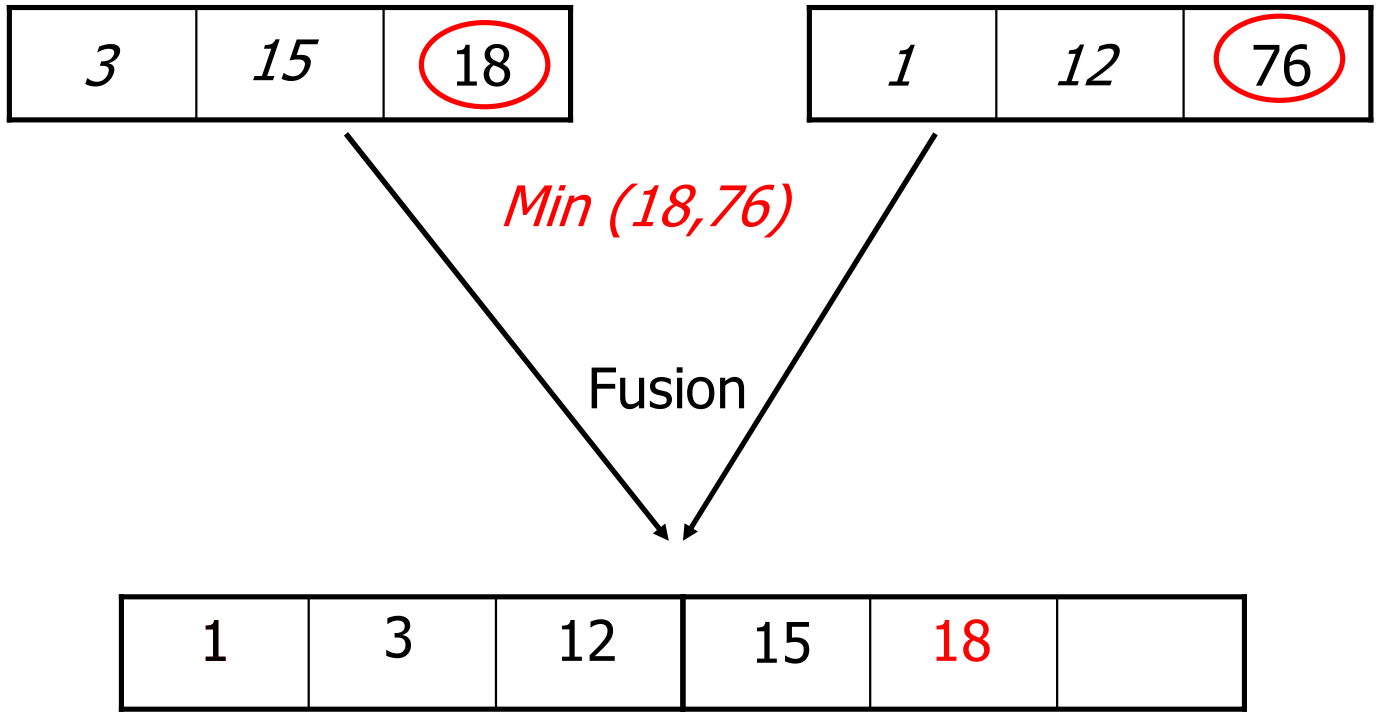
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



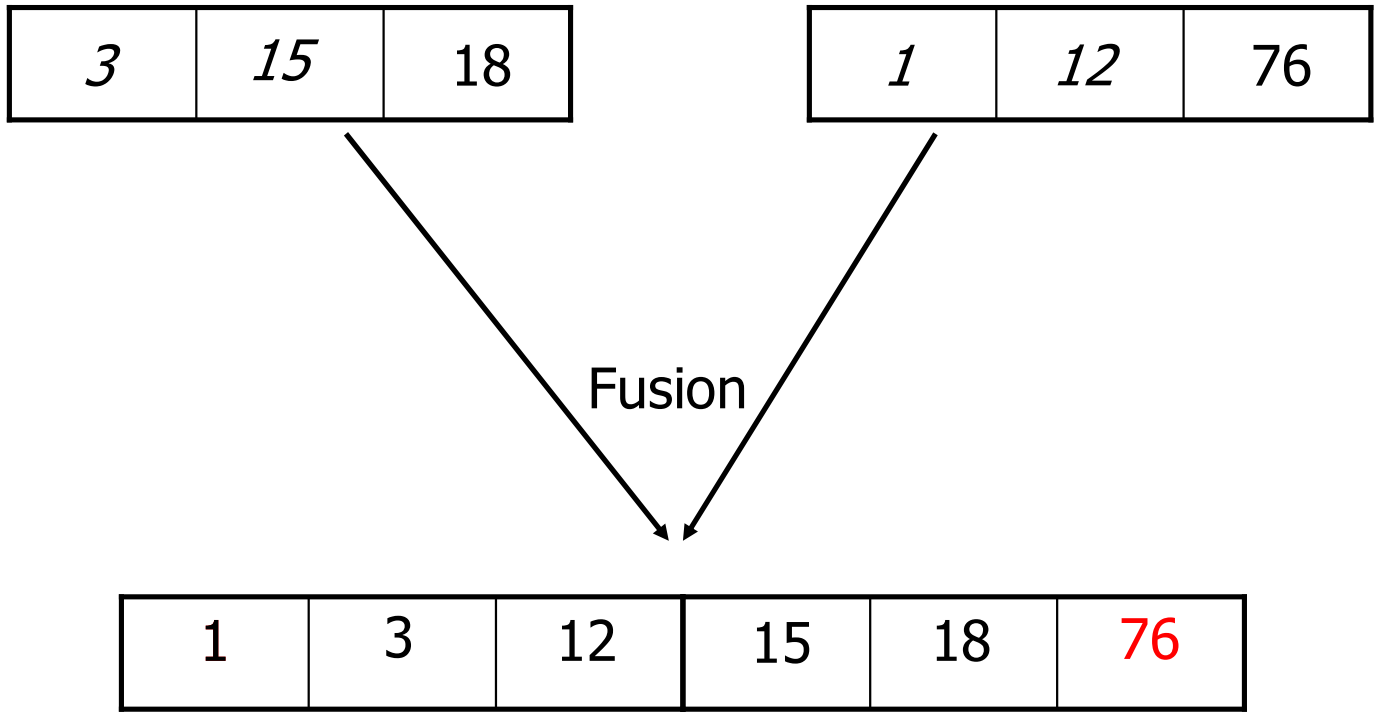
Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple



Méthodes algorithmiques pour la géométrie

- Division et fusion / Exemple
  - Analyse
    - Division s'effectue en temps linéaire
    - Fusion également en temps linéaire (à chaque pas on a 1 seule comparaison, et on progresse d'un élt)

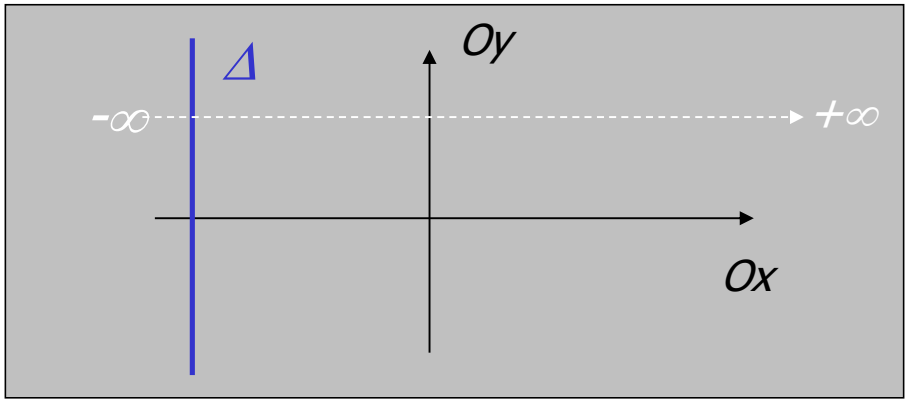
☞ $f(n) = \Theta(n)$
$p = 2$ et $q=2$ (2 sous problèmes de taille n/2)

D'après la théorie on est dans le cas $p = f(q) = 2$

D'où $t(n)= \Theta(n^1 \log n) = \Theta(n \log n)$

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage
  - Résout un problème 2D en simulant le balayage d'un plan par une droite



droite Δ parallèle à Oy balaye le plan qd elle se déplace continûment de gauche à droite

de la position initiale x= -∞ à la position finale d'abscisse x = + ∞

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage
  - Algos qui mettent en œuvre un balayage sont assez différents les uns des autres mais utilisent tous 2 structures de données
    - Une structure *Y* qui stocke les informations relatives à l'état du balayage
      - Informations dépendent du problème traité mais les propriétés suivantes sont toujours vérifiées
        - Infos contenues dans la structure *Y* sont liées à la position de la droite de balayage et évoluent lorsque celle-ci se déplace
        - La structure *Y* ne doit être mise à jour que lorsque la droite de balayage passe par un nombre fini de positions discrètes appelées *évènements*
        - *Le fait de maintenir cette structure tout au long du balayage permet à l'algo de construire la solution du problème*

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite)
  - Une structure *X* qui contient la suite des événements à traiter
    - Suite peut être entièrement connue au départ
    - Ou découverte au fur et à mesure que le balayage progresse
  - Initialisation de l'algo
    - Initialisation de la structure *Y* pour la position x= $-\infty$ de la droite de balayage
    - Initialisation de la structure *X* avec la suite, ordonnée suivant les abscisses croissantes des évènements connus au départ
  - A chaque évènement traité
    - Structure Y est mise à jour
    - Nouveaux évènements sont détectés et inclus dans *X* ou d'autres sont supprimés de *X*

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite)
  - Structures de données utilisées
    - Structure X
      - Si tous les évènements sont connus au départ ⇔ *simple liste chaînée*
      - Si évènements sont découverts au cours du balayage => structure doit gérer opérations de minimum, recherche , insertion (voire suppression) ⇔ *Queue de priorité*
    - Structure Y
      - Composantes doivent gérer un ens d'objets totalement ordonné permettant les opérations de recherche, insertion, suppression, et parfois prédécesseur, successeur

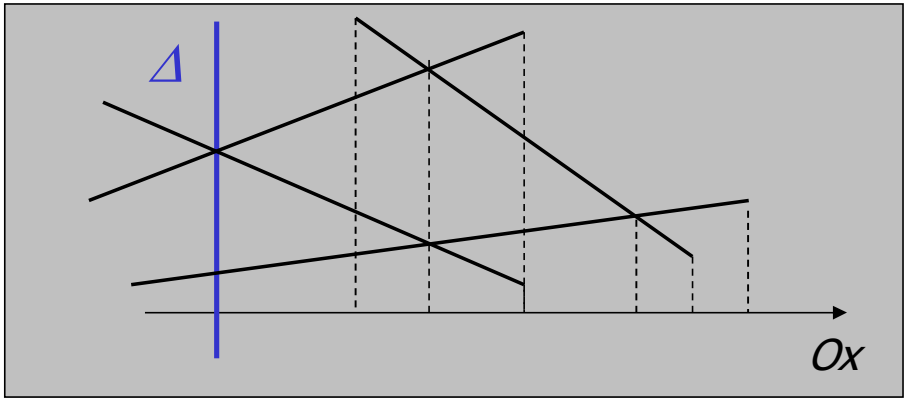
      ⇔ *Dictionnaire ou Dictionnaire augmenté*
  - *Remarque*
    - *Méthode qui peut se généraliser en dimension 3 (hyperplan de balayage)*

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Intersection de segments
    - *S* est un ensemble de n segments du plan
    - Problème : détecter toutes les paires de segments de *S* qui s'intersectent et calculer les coordonnées de leurs points d'intersection
    - Algorithme naïf : tester chacune des n(n-1)/2 paires de segments
      - ☞ Complexité $\Theta(n^2)$
      - ☞ Or le nb a d'intersections est largement inférieur à n^2
      - ☞ Plus intéressant de disposer d'1 algo dont la complexité est fonction également de la taille de la sortie

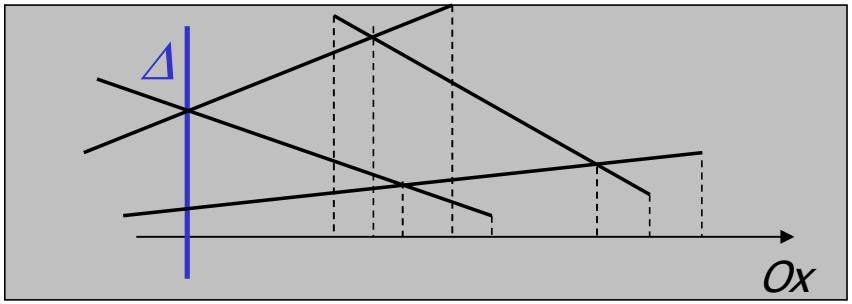
Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman
    - Algorithme en O((*n*+*a*) log *n*) pour un ens de *n* segments présentant *a* points d'intersection



Méthodes algorithmiques pour la géométrie

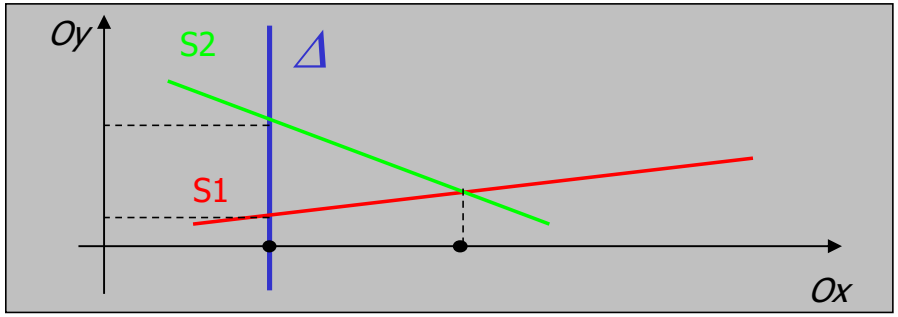
- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman(suite)
    - Hypothèse de situation générale pour simplifier la description
      - 3 segments quelconques de *S* n'ont pas d'intersection commune
      - Extrémités des segments de *S* et les pts d'intersection de *S* ont des abscisses toutes distinctes => *S* ne contient pas de segment vertical



    - Cas où cette hypothèse générale n'est pas vérifiée sont des cas particuliers dont le traitement est facile (si même ordre sur Ox => on prend ordre sur Oy)

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman (suite)
    - Idée de base :
      - si 2 segments S1 et S2 de *S* s'intersectent , alors toute droite Δ dont l'abscisse est suffisamment proche de celle de S1 ∩ S2, intersecte S1 et S2
      - De plus, S1 et S2 sont consécutifs dans la suite des segments de *S* intersectés par Δ (triés par ordre croissant des ordonnées de leur point d'intersection avec Δ)



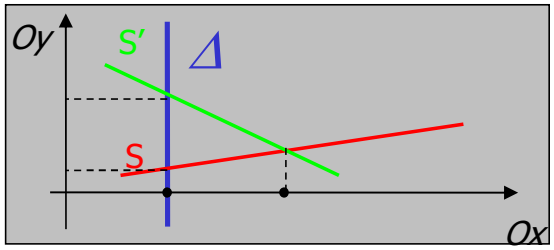
S1 et S2 sont des segments actifs car ils intersectent Δ

S1 : précédent de S2
S2 : suivant de S1

☞ 1 pt d'intersection I est découvert quand 2 segments S1 et S2 deviennent consécutifs dans la suite des segments actifs

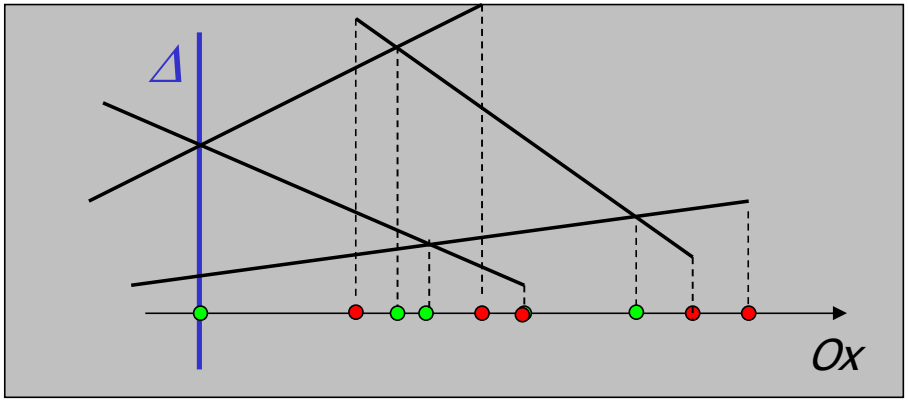
Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman (suite)
    - La structure *Y* maintient la liste des segments actifs ordonnés
    - Elle est modifiée quand Δ rencontre l'une des extrémités d'un segment de *S* ou un pt d'intersection
      - Si pt rencontré = extrémité gauche d'un segment
        - ☞ le segment doit être inséré dans *Y*
      - Si pt rencontré = extrémité droite d'un segment
        - ☞ le segment doit être retiré de *Y*
      - Si pt rencontré = pt d'intersection de S et S'
        - ☞ S et S' échangent leur place dans *Y*



Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman (suite)
    - La structure *X* des évènements à traiter inclut
      - L'ensemble des points extrémités des segments (connu au départ)
      - L'ensemble des points d'intersection (inconnu)



- Evt présent dans *X pour la position de Δ donnée*
- Evt qui vont être introduits dans *X*

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman (suite)
    - Initialisation
      - *X* initialisée avec la suite des extrémités des segments de *S* triée par ordre d'abscisses croissantes
      - *Y* est vide
    - Puis tant que *X* n'est pas vide
      - Extraction de *X* de l'évt d'abscisse minimale Evt $_{min}$
      - Traitement de Evt $_{min}$:
        - Cas où evt = extrémité gauche d'un segment S

Cas 1

=> S inséré dans *Y*

=> Si le successeur (ou le précédent) de S dans *Y* et S s'intersectent, leur point d'intersection est calculé et inséré dans *X*

        - Cas où evt = extrémité droite E d'un segment S

Cas 2

=> S supprimé de *Y*

=> Si le successeur et le précédent de S dans *Y* s'intersectent à droite de E, leur point d'intersection est inséré dans *X (on vérifie qu'il n'existe pas déjà dans X)*

        - Cas où evt = pt d'intersection I de 2 segments S et S'

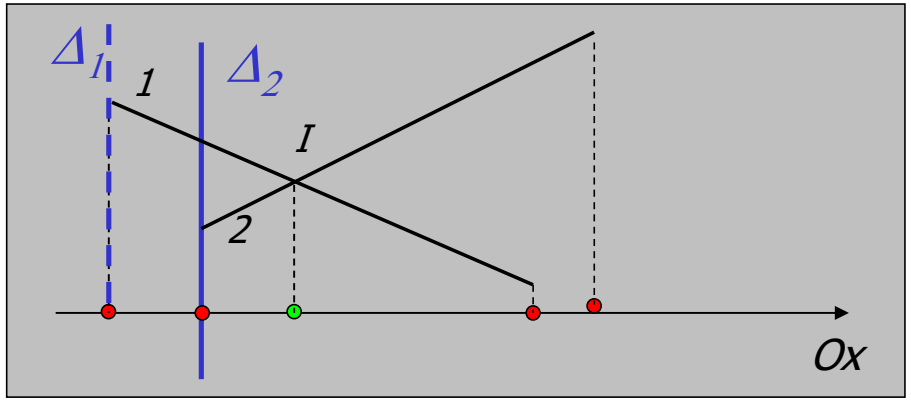
Cas 3

=> S et S' sont échangés dans Y

=> Si S précède S', on teste les intersections entre S et pred(S) et S' et suiv(S'), et tout pt d'intersection trouvé (abscisse > abscisse(I)) est inséré s'il n'est pas déjà présent

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman: Cas 1 => segment 2 est inséré



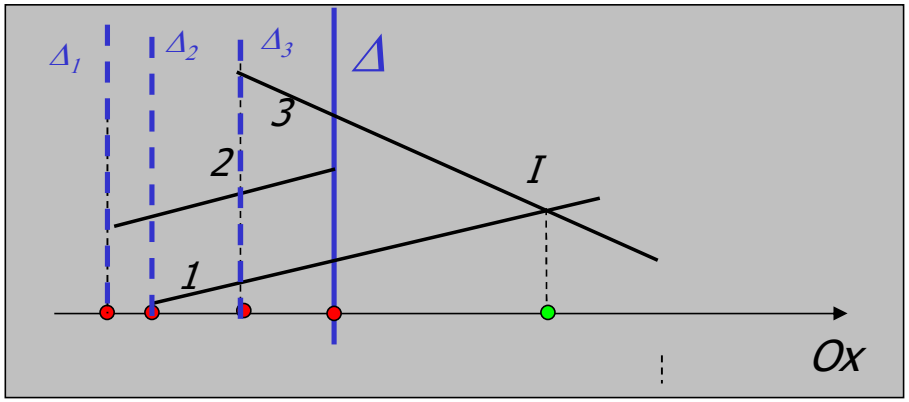
Δ_1 Δ_2

1 *(suiv de 2)* et pt I d'intersection entre 2 et son suivant (1) est calculé et inséré dans la liste *X* selon son abscisse

1 *2* *(pred de 1)*

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman: Cas 2 => on retire segment 2



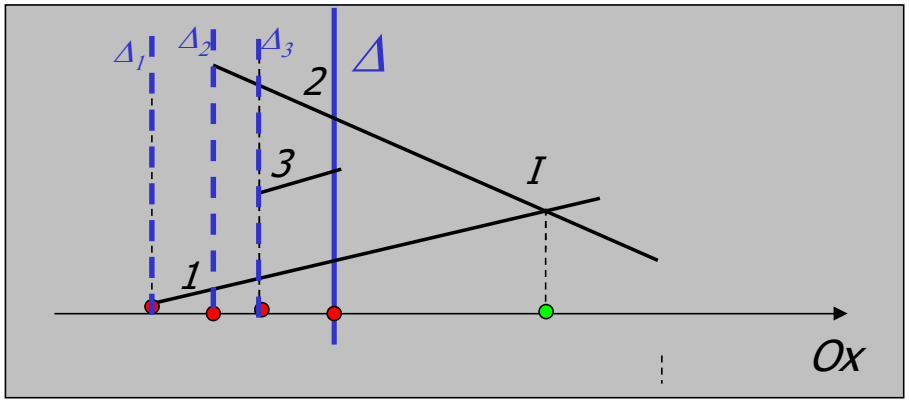
Δ_1	Δ_2	Δ_3	Δ	
		3		
	2	2	3	*Suppression de 2 et Test d'intersection entre 3 (suiv 2) et 1(pred(2)) => Insertion de I*
2	1	1	1	

Test d'intersection entre 2 et 1 (pred(2)) mais pas entre 2 et son suivant qui n'existe pas

Test d'intersection entre 3 et 2 (pred(3)) mais pas entre 3 et son suivant qui n'existe pas

Méthodes algorithmiques pour la géométrie

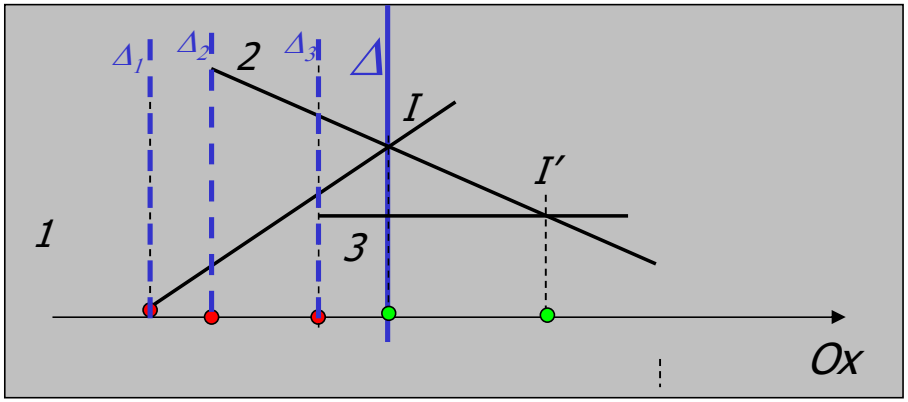
- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman: Cas 2 => on retire segment 3



Δ_1	Δ_2		Δ_3		Δ	
	2	*Test d'intersection entre 2 et 1 (pred(2)) mais pas entre 2 et son suivant qui n'existe pas => pt I inséré*	2	*Pas d'inter entre 1 et 3 Ni entre 3 et 2*	2	*Suppression de 3 et Test d'intersection entre 2 (suiv 3) et 1(pred(3)) => Calcul de I qui existe déjà donc non inséré*
1	1		3		1	
			1			

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman: Cas 3 => on inverse segments 1 et 2



Δ_1 Δ_2

2 — *Test d'intersection entre 2 et 1 (pred(2)) => pt I inséré*

1 1

Δ_3

2

1 — *Pas d'inter entre 3 et 1 Pred de 3 n'existe pas*

3

Δ

1

2 — *Inversion de 1 et 2 On teste intersection entre 3 et 2 => pt I' détecté et inséré*

3

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman: Analyse
    - Structure de données *Y*
      - contient au max O(n) segments
      - Si Dictionnaire augmenté représenté par 1 arbre équilibré
        - ☞ chaque opération (insertion, suppression, recherche) est effectuée en O(log n)
        - ☞ Opérations successeur et prédécesseur effectuées en temps constant
      - ☞ Structure *Y* peut être traitée en O(log n)
    - Structure de données *X*
      - contient au max O(n+a) évènements
      - Si queue de priorité représentée par 1 arbre équilibré
        - ☞ chaque opération (insertion, suppression, recherche, minimum) est effectuée en O(log (n+a)) donc O(log(n))
        - ☞ Opérations successeur et prédécesseur effectuées en temps constant
      - ☞ Structure *X* peut être traitée en O(log n)

Méthodes algorithmiques pour la géométrie

- Algorithmes de Balayage (suite) / Exemple
  - Algo de Bentley-Ottman: Analyse

  D'où
    - Étape initiale qui trie les 2n abscisses des extrémités des segments et initialise la structure *X* => O(2n * log(n)) = O(n log(n)) opérations élémentaires
    - Ensuite 2n + a évènements sont traités
      - Chaque évt implique un nb borné d'opérations dans chacune des structures X et Y => O(log n)
      - O((2n+a) log n)
    - Complexité totale en temps de calcul O((n+a) log(n))
      en mémoire O (n+a)

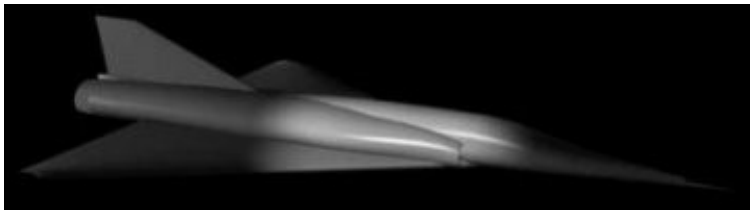
SOMMAIRE

- Informations pratiques
- Introduction
- Notions Algorithmiques
- Méthodes Algorithmiques pour la géométrie
- Modéliser *le monde*
- Méthodes géométriques
  - Notions de base en géométrie
  - Méthodes applicables aux modèles discrets
  - Méthodes applicables aux modèles continus

Représenter un objet / étapes

- la modélisation (créer des formes sans couleurs)
- le texturage (mettre des couleurs + textures)

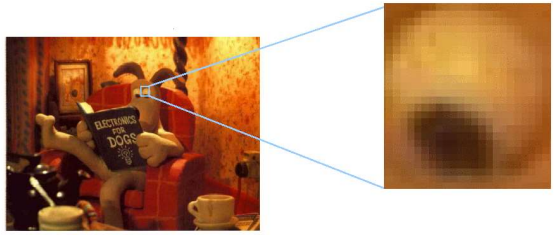
- le positionnement des lumières

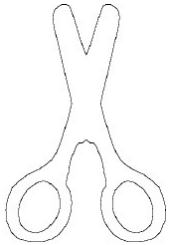


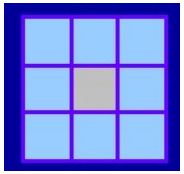
- le rendu
  - Méthode qui permet de transformer les coordonnées et paramètres du modèle en 1 ou plusieurs images (ex: lancer de rayon)

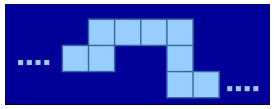
Modélisation d'1 scène ou 1 objet

- Monde discret
  - Manipulation de données géométriques discrètes pour représenter un phénomène continu
  - Pixels (2D)

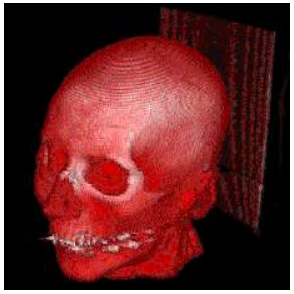


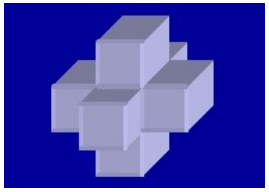


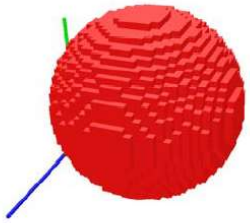




  - Voxels (3D)

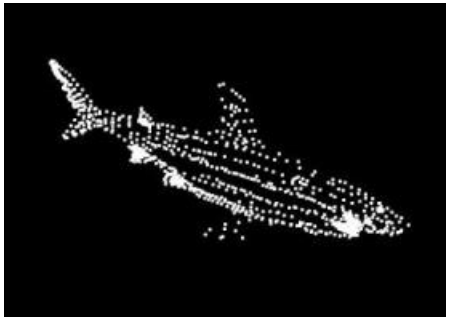






Modélisation d'1 scène ou 1 objet

- Monde discret
  - Nuage de points (x,y,z)
    - Pb de l'échantillonnage (nb de points)
    - Pb de la répartition des points



  - Pb des représentations discrètes
    - absence de structuration de l'information, présence de bruit

☞Nécessité d'utiliser des méthodes qui vont structurer l'information en s'affranchissant du bruit

  - Méthodes employées sont liées à la géométrie discrète

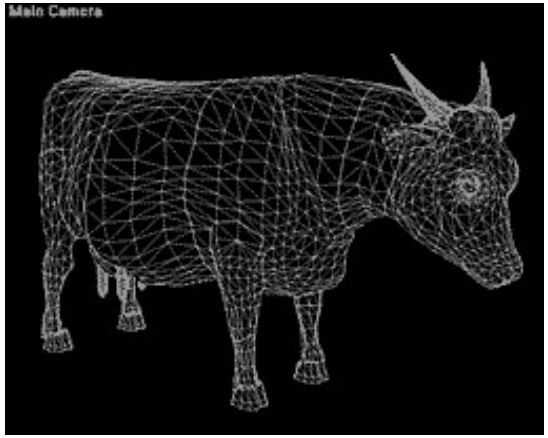
(Spécification discrète de la géométrie euclidienne)

Modélisation d'1 scène ou 1 objet

- Modéliser un objet ou une scène avec des modèles continus
  - Utiliser un ens de primitives ou de formes géométriques
    - assez simples => pour une implémentation facile
    - assez souples => pour modéliser une grande variété d'objets
  - Primitives utilisables par ordre de complexité croissante
    - Points
    - Segments
    - Lignes brisées
    - Polygones
    - Surfaces
    - Polyèdres

Modélisation d’1 scène ou 1 objet

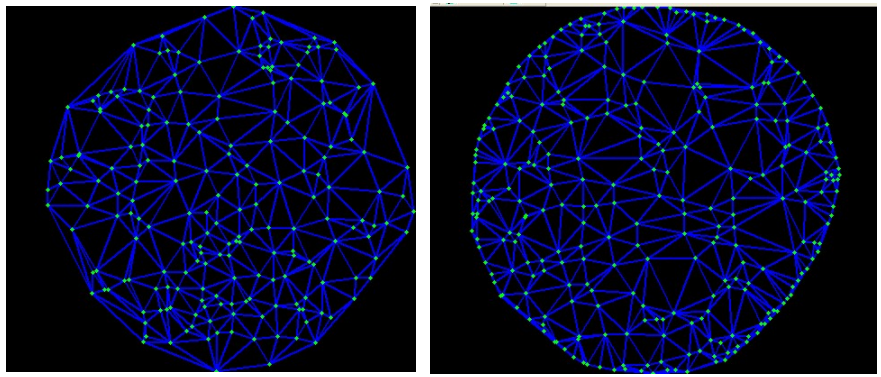
- 3 types de modélisation
  - Modélisation par fil de fer
    - Objet est décrit par ses arêtes
    - Avantages: modèle simple à construire et à manipuler

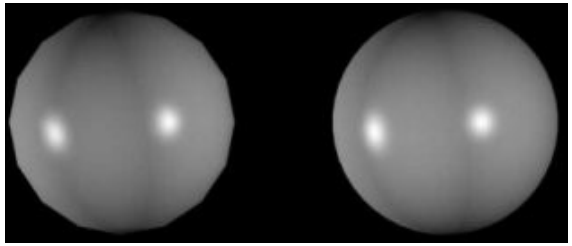


Modélisation d'1 scène ou 1 objet

- Modélisation par fil de fer (suite)
  - Pb: si modèle à construire comporte des fortes courbures

☞ Il faut employer de très nombreux polygones si on veut une bonne impression d'arrondi et de lissage

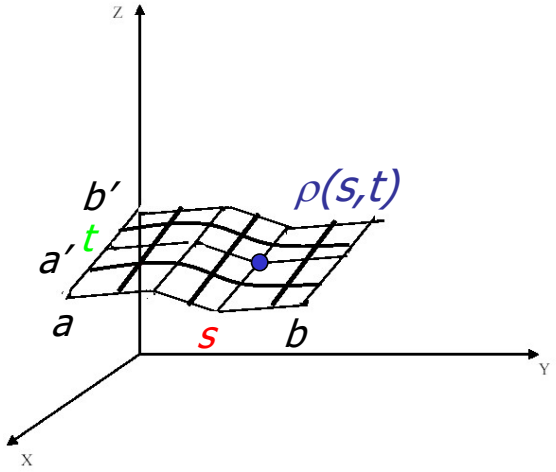


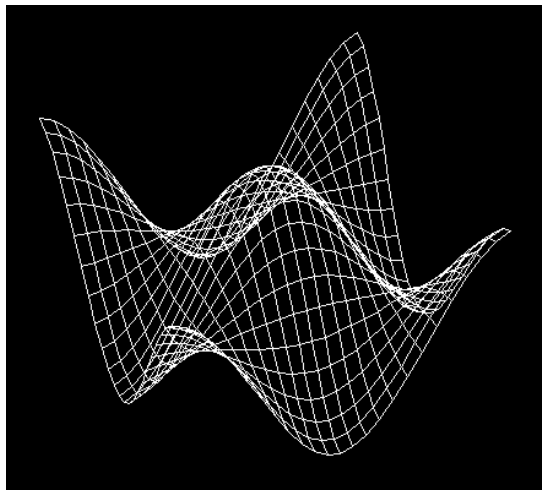


224 faces *2024 faces*

Modélisation d'1 scène ou 1 objet

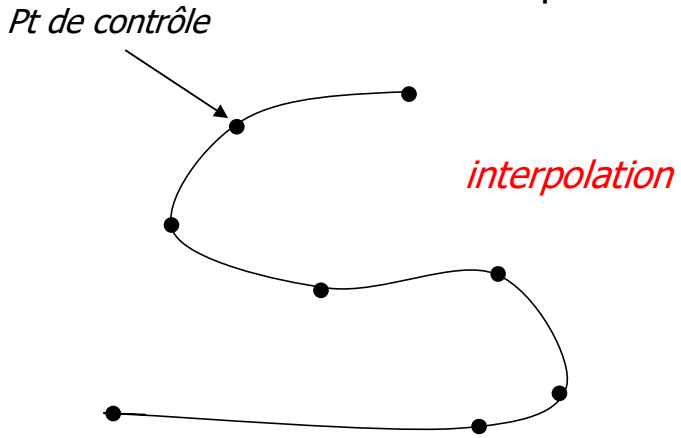
- 3 types de modélisation (suite)
  - Modélisation surfacique paramétrique
    - Objet représenté par les surfaces frontières
    - Utilisation des courbes ou surfaces paramétrées (B-splines, NURBS, courbes de Bézier) => surfaces définies par des équations

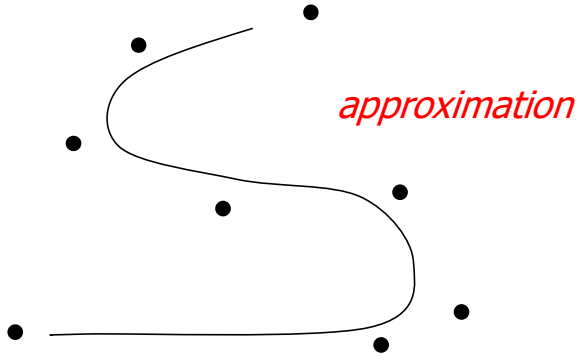


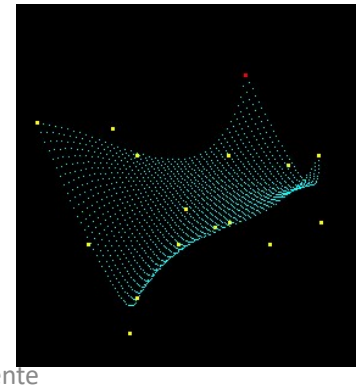


Modélisation d'1 scène ou 1 objet

- Courbes ou surfaces paramétrées
  - Pb revient à trouver l'équation de la courbe ou de la surface qui "*passe*" par un ensemble de points de contrôle

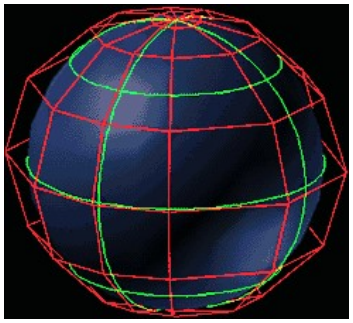






+ le nb de points de contrôle est élevé
⇒Plus la représentation sera précise
⇒Plus la complexité de la méthode de calcul sera importante

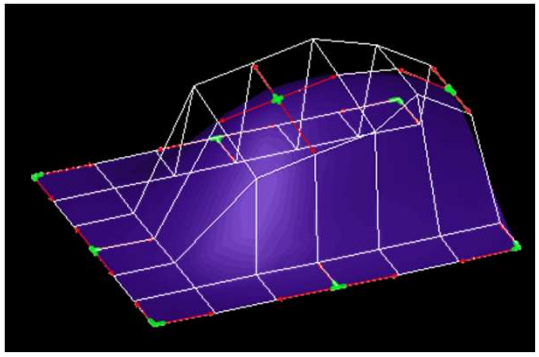
Modélisation d’1 scène ou 1 objet



NURBS (B-splines non uniformes rationnelles)

-les courbes qui mettent en évidence la surface (en vert)
- les lignes de contrôle qui relient les points de contrôle (en rouge)

Chaque pt de contrôle est affecté d’un poids (plus le poids est élevé plus la courbe passe près du point de contrôle)



Courbes de Bézier

- les points de contrôle (en vert)
- les vecteurs force (flèches rouges)
- les lignes de contrôle (en blanc)

qui permettent de prévoir la surface

Modélisation d’1 scène ou 1 objet

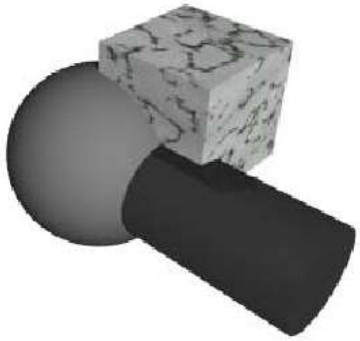
- Modélisation surfacique paramétrique (suite)
  - Haut niveau de modélisation => élaboration de modèles très réalistes
    - Ces courbes ont l’avantage de ne pas "facetter"

    ☞bords toujours lisses quel que soit le niveau de zoom utilisé (pas le cas des modèles en fil de fer)

    - Pb de continuité peut se poser au niveau des raccords des morceaux de courbes
  - Modèles éditables inter-activement
    - Très utilisés pour le design , conception mécanique ….

Modélisation d'1 scène ou 1 objet

- 3 types de modélisation (suite)
  - Modélisation volumique
    - Notion de solide et de matière sont incluses
    - Localisation des zones vides ou pleines
    - Une des méthodes : géométrie constructive (CSG)
      - Solide complexe composés de solides + simples



La surface.
On entend par surface tout ce qui enveloppe les corps; ne pas trop se fier aux surfaces, elles sont souvent trompeuses.

Modélisation d’1 scène ou 1 objet

- Quel type de modélisation choisir ?
  - Cela dépend
    - Des données
    - Des informations liées aux données ou au modèle
    - Des traitements utilisés
    - Des résultats attendus