# Algorithme 1
# Diviser pour régner

Hajer Akid & Sandrine Julia

Semestre 4



## Diviser pour régner

- Diviser un problème en sous-problèmes qui sont des instances plus petites du même problème
- Résoudre récursivement ces sous-problèmes
- Combiner ces réponses pour obtenir la solution au problème

**Avantages**

- Simplification conceptuelle
- Accélération
  - réduire rapidement (de manière exponentielle) l'espace des problèmes
  - exploiter les points communs dans les solutions des sous-problèmes
- Parallélisme
- Localité



## Diviser pour régner

**Exemples**

- Recherche dichotomique
- Tri fusion
- Karatsuba



## Le problème

- Bien **comprendre** le problème
- Faire le lien entre l'entrée et la sortie
- Utiliser des bâtons pour représenter l'entrée

  [|||||||||||||] les éléments d'indice 1 à $n$

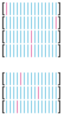
  Les éléments peuvent être
  - triés ou non
  - dans un tableau, une pile, un tas, une liste
  - des nombres, des entiers d'un nombre, des éléments d'un ensemble

## L'élément fil rouge

- Bien **comprendre** à quoi correspondent les bâtons
- Trouver l'élément fil rouge permettant de formuler les sous-problèmes
  1. le premier élément
  2. le dernier élément
  3. l'élément du milieu
  4. la racine si c'est un arbre

  ...

  5. élément optimal
  6. un élément aléatoire
  7. un pivot





## L'élément fil rouge

- L'élément fil rouge permet-il d'étendre une solution partielle ?
- L'élément fil rouge permet-il de formuler les sous-problèmes ?
- Quelle est la complexité de la combinaison de l'élément filrouge avec leur sous-problèmes ou solutions partielles ?



## Recherche dichotomique

Trouver un élément *a* dans un tableau trié de *n* cases

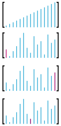
```
BS (L,i,j,a){
  si (j < i) { retourner FAUX }
  tt ← i + ⌊(j−i)/2⌋
  si (L[tt] < a) {
    retourner BS (L,tt+1,j,a)
  }
  sinon si (L[tt] > a ) {
    retourner BS (L,i,tt−1,a)
  }
  sinon { retourner VRAI }
}
```



## Recherche

Trouver un élément *a* dans un tableau de *n* cases

- Tri
- Premier élément fil rouge
- Dernier élément fil rouge
- Élément du milieu fil rouge



## Exponentiation

Calculer $x^n$ avec $n \geq 0$

```
puissance (x, n) {
  y ← 1
  pour (i de 1 à n) {
    y ← x * y
  }
  retourner y
}
```



## Exponentiation rapide

Calculer $x^n$ avec $n \geq 0$

```
puissance (x, n) {
  si (n > 1) {
    y ← puissance (x*x, ⌊n/2⌋)
    si (n mod 2 = 1) {
      y ← x * y
    }
    retourner y
  }
  retourner x
}
```



## Tri fusion

L'algorithme peut être décrit récursivement

- On découpe en deux parties à peu près égales les données à trier
- On trie les données de chaque partie
- On fusionne les deux parties

La récursivité s'arrête car on finit par arriver à des listes composées d'un seul élément et le tri est alors trivial



## Tri fusion

Fusion de deux tableaux triés

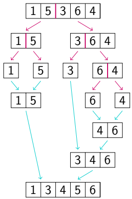
| 1 | 3 | 5 |
|---|---|---|

| 2 | 7 | 9 |
|---|---|---|

| 1 | 2 | 3 | 5 | 7 | 9 |
|---|---|---|---|---|---|

## Tri fusion

### Tri dichotomique





## Tri fusion

```
merge_sort (L, i, j) {
  si (j − i > 0) {
    tt ← i + ⌊(j−i)/ 2⌋
    merge_sort (L, i, tt)
    merge_sort (L, tt+1, j)
    merge (L, i, j)
  }
}
```



## Multiplication de matrices carrées

Soient $X$ et $Y$ deux matrices de $n \times n$
Calculer $X \cdot Y = Z$
$Z_{ij} = \sum_{k=1}^{n} X_{ik} \cdot Y_{kj}$

### Exemple

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}$$



## Multiplication de matrices carrées

### Idée

Découper $X$ et $Y$ en matrices carrées plus petites

$$X = \left(\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right), \ Y = \left(\begin{array}{c|c} E & F \\ \hline G & H \end{array}\right)$$

$$X \cdot Y = \left(\begin{array}{c|c} AE + BG & AF + BH \\ \hline CE + DG & CF + DH \end{array}\right)$$

$T(n) = 8T\left(\frac{n}{2}\right) + n^2$
$T(n) \in O(n^3)$

## Algorithme de Strassen (1969)

7 multiplications !

$T(n) = 7T\left(\frac{n}{2}\right) + n^2$
$T(n) \in O(n^{\log_2(7)})$

$M_1 = A(F - H)$
$M_2 = (A + B)H$
$M_3 = (C + D)E$
$M_4 = D(G - E)$
$M_5 = (A + D)(E + H)$
$M_6 = (B - D)(G + H)$
$M_7 = (C - A)(E + F)$

$$X \cdot Y = \left(\begin{array}{c|c} AE + BG & AF + BH \\ \hline CE + DG & CF + DH \end{array}\right)$$



## Algorithme de Strassen (1969)

7 multiplications !

$T(n) = 7T\left(\frac{n}{2}\right) + n^2$
$T(n) \in O(n^{\log_2(7)})$

$M_1 = A(F - H)$
$M_2 = (A + B)H$
$M_3 = (C + D)E$
$M_4 = D(G - E)$
$M_5 = (A + D)(E + H)$
$M_6 = (B - D)(G + H)$
$M_7 = (C - A)(E + F)$

$$X \cdot Y = \left(\begin{array}{c|c} M_5 + M_4 - M_2 + M_6 & M_1 + M_2 \\ \hline M_3 + M_4 & M_7 + M_1 - M_3 + M_5 \end{array}\right)$$



## Deux points les plus rapprochés

Soit $P = \{p_1, \ldots, p_n\}$ un ensemble de points
On note $d(p_i, p_j)$ la distance euclidienne entre $p_i = (x_i, y_i)$ et $p_j = (x_j, y_j)$

$$d(p_i, p_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Trouver deux points $p_a, p_b \in P$ qui minimisent $d(p_a, p_b)$
On suppose que tous les points ont des coordonnées distinctes



## Deux points les plus rapprochés

En une dimension

Soit $P = \{p_1, \ldots, p_n\}$ un ensemble de points
On note $d(p_i, p_j)$ la distance euclidienne entre $p_i = (x_i)$ et $p_j = (x_j)$

$$d(p_i, p_j) = |x_i - x_j|$$

Trouver deux points $p_a, p_b \in P$ qui minimisent $d(p_a, p_b)$