

# Tris de tableaux

*Du quadratique au ennelogue*

$\psi$

stephane@gonnord.org - <http://blog.psi945.fr>

Lundi 17 octobre, 7, 14 et 21 novembre 2016



# Plan

- 1 Un peu de contexte
  - De tout temps...
  - Ce qui/qu'on compte
  - Tour d'horizon
- 2 Tris quadratique
  - Tri bulle
  - Sélection
  - Insertion
- 3 Divide and conquer : deux tris récursifs
  - Tri fusion
  - Tri rapide

# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- Tri bulle
- Sélection
- Insertion

## 3 Divide and conquer : deux tris récursifs

- Tri fusion
- Tri rapide

# De tout temps... l'homme a voulu trier/classer

- Étape dans d'autres algorithmes.

# De tout temps... l'homme a voulu trier/classer

- Étape dans d'autres algorithmes.
- Cas d'école en algorithmique/complexité, pour le jeune padawan.

# De tout temps... l'homme a voulu trier/classer

- Étape dans d'autres algorithmes.
- Cas d'école en algorithmique/complexité, pour le jeune padawan.
- Ressources disponibles : temps, espace, processeurs.

# Sommaire

- 1 Un peu de contexte
  - De tout temps...
  - Ce qui/qu'on compte**
  - Tour d'horizon

- 2 Tris quadratique
  - Tri bulle
  - Sélection
  - Insertion

- 3 Divide and conquer : deux tris récursifs
  - Tri fusion
  - Tri rapide

# Ce qui/qu'on compte

- Comparaisons, affectations/échanges dans le tableau.



# Ce qui/qu'on compte

- Comparaisons, affectations/échanges dans le tableau.
- Dans le pire (meilleur) des cas, « en moyenne ».

# Ce qui/qu'on compte

- Comparaisons, affectations/échanges dans le tableau.
- Dans le pire (meilleur) des cas, « en moyenne ».
- Constantes multiplicatives.

# Ce qui/qu'on compte

- Comparaisons, affectations/échanges dans le tableau.
- Dans le pire (meilleur) des cas, « en moyenne ».
- Constantes multiplicatives.
- En place ou non ?

# Ce qui/qu'on compte

- Comparaisons, affectations/échanges dans le tableau.
- Dans le pire (meilleur) des cas, « en moyenne ».
- Constantes multiplicatives.
- En place ou non ?
- Parallélisable ou non ?

# Une borne inférieure théorique

## Théorème

On ne peut pas trier de tableau de taille  $n$  en moins de  $n \ln n$  comparaisons.

# Une borne inférieure théorique

## Théorème

On ne peut pas trier de tableau de taille  $n$  en moins de  $n \ln n$  comparaisons.

- C'est faux ! Tableau de  $n$  bits ou  $n$  pixels de  $[0, 255]$ ...

# Une borne inférieure théorique

## Théorème

On ne peut pas trier de tableau de taille  $n$  en moins de  $n \ln n$  comparaisons.

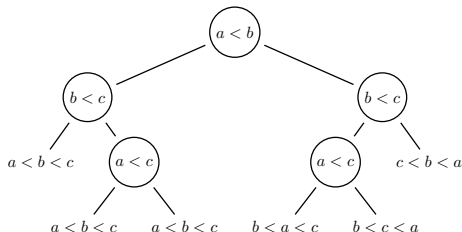
- C'est faux ! Tableau de  $n$  bits ou  $n$  pixels de  $[0, 255]$ ...
- C'est vrai... modulo quelques hypothèses.

# Une borne inférieure théorique

## Théorème

On ne peut pas trier de tableau de taille  $n$  en moins de  $n \ln n$  comparaisons.

- C'est faux ! Tableau de  $n$  bits ou  $n$  pixels de  $[0, 255]$ ...
- C'est vrai... modulo quelques hypothèses.
- Preuve ? Via des arbres de décision : pas pour nous !





# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- Tri bulle
- Sélection
- Insertion

## 3 Divide and conquer : deux tris récursifs

- Tri fusion
- Tri rapide

# Tour d'horizon

Complexité des algorithmes classiques :

- Sélection, insertion :  $n^2$ ...

# Tour d'horizon

Complexité des algorithmes classiques :

- Sélection, insertion :  $n^2$ ... mais on finassera.
- Tri fusion, tri rapide :  $n \ln n$ .

# Tour d'horizon

Complexité des algorithmes classiques :

- Sélection, insertion :  $n^2$ ... mais on finassera.
- Tri fusion, tri rapide :  $n \ln n$ .
- Mieux ?

# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- **Tri bulle**
- Sélection
- Insertion

## 3 Divide and conquer : deux tris récursifs

- Tri fusion
- Tri rapide

# Tri bulle

**Entrées :**  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire

**si**  $T[i] > T[i+1]$  **alors**

$T[i] \leftrightarrow T[i+1]$

10	40	7	30	20	15
----	----	---	----	----	----

# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

10	40	7	30	20	15
----	----	---	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

10	40	7	30	20	15
----	----	---	----	----	----





# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

10	7	40	30	20	15
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  alors

└  $T[i] \leftrightarrow T[i+1]$

└

10	7	40	30	20	15
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

10	7	30	40	20	15
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

10	7	30	40	20	15
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

10	7	30	20	40	15
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

10	7	30	20	40	15
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à 1 **faire**

┌ **pour**  $i$  de 0 à  $j-1$  **faire**  
├ ┌ **si**  $T[i] > T[i+1]$  **alors**  
└ └ ┌  $T[i] \leftrightarrow T[i+1]$

10	7	30	20	15	40
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

10	7	30	20	15	40
----	---	----	----	----	----





# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

10	7	30	20	15	40
----	---	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

7	10	30	20	15	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

7	10	30	20	15	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	30	20	15	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	20	30	15	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

7	10	20	30	15	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à 1 **faire**

```
┌   pour  $i$  de 0 à  $j-1$  faire
├       si  $T[i] > T[i+1]$  alors
├            $T[i] \leftrightarrow T[i+1]$ 
└
```

7	10	20	15	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	20	15	30	40
---	----	----	----	----	----





# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

7	10	20	15	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

7	10	20	15	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

```
┌   pour  $i$  de  $0$  à  $j-1$  faire
├     si  $T[i] > T[i+1]$  alors
├        $T[i] \leftrightarrow T[i+1]$ 
└
```

7	10	20	15	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

```
┌   pour  $i$  de  $0$  à  $j-1$  faire
├       si  $T[i] > T[i+1]$  alors
├            $T[i] \leftrightarrow T[i+1]$ 
└
```

7	10	15	20	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	15	20	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

┌  $T[i] \leftrightarrow T[i+1]$

└

└

└

7	10	15	20	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	15	20	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  **alors**

└  $T[i] \leftrightarrow T[i+1]$

└

7	10	15	20	30	40
---	----	----	----	----	----





# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

┌ **pour**  $i$  de  $0$  à  $j-1$  faire

┌ **si**  $T[i] > T[i+1]$  alors

┌  $T[i] \leftrightarrow T[i+1]$

7	10	15	20	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à  $1$  faire

**pour**  $i$  de  $0$  à  $j-1$  faire  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	15	20	30	40
---	----	----	----	----	----



# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à 1 **faire**

```
┌   pour  $i$  de 0 à  $j-1$  faire
├       si  $T[i] > T[i+1]$  alors
├            $T[i] \leftrightarrow T[i+1]$ 
└
```

7	10	15	20	30	40
---	----	----	----	----	----

- Comparaisons :  $\frac{n(n-1)}{2} = n^2$ .
-

# Tri bulle

Entrées :  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n-1$  à 1 **faire**

**pour**  $i$  de 0 à  $j-1$  **faire**  
        **si**  $T[i] > T[i+1]$  **alors**  
             $T[i] \leftrightarrow T[i+1]$

7	10	15	20	30	40
---	----	----	----	----	----

- Comparaisons :  $\frac{n(n-1)}{2} = n^2$ .
- Échanges : de 0 à  $\frac{n(n-1)}{2}$  ;  $\frac{n(n-1)}{4}$  en moyenne.

# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- Tri bulle
- **Sélection**
- Insertion

## 3 Divide and conquer : deux tris récursifs

- Tri fusion
- Tri rapide

# Tri sélection

10	40	7	30	20	15
----	----	---	----	----	----

# Tri sélection

10	40	7	30	20	15
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ;

# Tri sélection

10	15	7	30	20	40
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ;  
le mettre à sa place finale.



# Tri sélection

10	15	7	30	20	40
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ;  
le mettre à sa place finale.
- Trouver la position du suivant ;

# Tri sélection

10	15	7	20	30	40
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.

# Tri sélection

10	15	7	20	30	40
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...

# Tri sélection

10	15	7	20	30	40
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...

# Tri sélection

10	15	7	20	30	40
----	----	---	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...

# Tri sélection

10	7	15	20	30	40
----	---	----	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...

# Tri sélection

10	7	15	20	30	40
----	---	----	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...

# Tri sélection

7	10	15	20	30	40
---	----	----	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...



# Tri sélection

7	10	15	20	30	40
---	----	----	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...

# Tri sélection

7	10	15	20	30	40
---	----	----	----	----	----

- Trouver le (la position du) plus gros élément ; le mettre à sa place finale.
- Trouver la position du suivant ; le mettre à sa place finale.
- ...
- Ou encore :

**Entrées :**  $T$

$n \leftarrow |T|$

**pour**  $j$  de  $n - 1$  à 1 **faire**

┌ Trouver la position  $p$  du maximum de  $T[:j+1]$   
└  $T[p] \leftrightarrow T[j]$

# Tri sélection

- Comment trouver la position du maximum de  $T[:j+1]$  ?

# Tri sélection

- Comment trouver la position du maximum de  $T[:j+1]$  ?

**Entrées** :  $T$  et  $j$

$p \leftarrow 0$

**pour**  $i$  de 1 à  $j$  **faire**

┌ **si**  $T[i] > T[p]$  **alors**  
└  $p \leftarrow i$

**Résultat** :  $p$

# Tri sélection

- Comment trouver la position du maximum de  $T[:j+1]$  ?

**Entrées :**  $T$  et  $j$

$p \leftarrow 0$

**pour**  $i$  de 1 à  $j$  **faire**

┌ **si**  $T[i] > T[p]$  **alors**  
└  $p \leftarrow i$

**Résultat :**  $p$

- Complexité :
  - ▶ Comparaisons :  $\frac{n(n-1)}{2}$

# Tri sélection

- Comment trouver la position du maximum de  $T[:j+1]$  ?

**Entrées** :  $T$  et  $j$

$p \leftarrow 0$

**pour**  $i$  de 1 à  $j$  **faire**

┌ **si**  $T[i] > T[p]$  **alors**  
└  $p \leftarrow i$

**Résultat** :  $p$

- Complexité :
  - ▶ Comparaisons :  $\frac{n(n-1)}{2}$
  - ▶ Échanges :  $n$

# Tri sélection

- Comment trouver la position du maximum de  $T[:j+1]$  ?

**Entrées :**  $T$  et  $j$

$p \leftarrow 0$

**pour**  $i$  de 1 à  $j$  **faire**

┌ **si**  $T[i] > T[p]$  **alors**  
└  $p \leftarrow i$

**Résultat :**  $p$

- Complexité :
  - ▶ Comparaisons :  $\frac{n(n-1)}{2}$
  - ▶ Échanges :  $n$
  - ▶ Cas où  $T$  est trié ?

# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- Tri bulle
- Sélection
- **Insertion**

## 3 Divide and conquer : deux tris récursifs

- Tri fusion
- Tri rapide



# Tri insertion

10	40	7	30	20	15
----	----	---	----	----	----

# Tri insertion

10	40	7	30	20	15
----	----	---	----	----	----

- On note que  $t[: 1]$  est bien trié ;

# Tri insertion

10	40	7	30	20	15
----	----	---	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ;

# Tri insertion

10	40	7	30	20	15
----	----	---	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;

# Tri insertion

10	40	7	30	20	15
----	----	---	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ;

# Tri insertion

7	10	40	30	20	15
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...

# Tri insertion

7	10	40	30	20	15
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...

# Tri insertion

7	10	30	40	20	15
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...



# Tri insertion

7	10	30	40	20	15
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...

# Tri insertion

7	10	20	30	40	15
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...

# Tri insertion

7	10	20	30	40	15
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...

# Tri insertion

7	10	15	20	30	40
---	----	----	----	----	----

- On note que  $t[: 1]$  est bien trié ;
- chercher la position de  $t[1]$  dans  $t[: 2]$  ; déplacer  $t[1]$  ;
- chercher la position de  $t[2]$  dans  $t[: 3]$  ; déplacer  $t[1]$  ;
- ...

- Ou encore :

**Entrées :**  $T$

$n \leftarrow |T|$

**pour**  $i$  de 1 à  $n - 1$  **faire**

$j \leftarrow i - 1$

**tant que**  $j \geq 0$  et  $T[j] > T[j + 1]$  **faire**

$T[j] \leftrightarrow T[j + 1]$

$j \leftarrow j - 1$

# Tri insertion : complexité

- En place ; on ne retourne rien.

# Tri insertion : complexité

- En place ; on ne retourne rien.
- Importance de l'évaluation paresseuse.

# Tri insertion : complexité

- En place ; on ne retourne rien.
- Importance de l'évaluation paresseuse.
- Version slicée.

# Tri insertion : complexité

- En place ; on ne retourne rien.
- Importance de l'évaluation paresseuse.
- Version slicée.
- Entre  $n$  et  $n^2$  comparaisons, autant d'échanges.



# Tri insertion : recherche dichotomique

- On trouve la position de  $T[i]$  en  $\ln_2 i$  comparaisons.

# Tri insertion : recherche dichotomique

- On trouve la position de  $T[i]$  en  $\ln_2 i$  comparaisons.
- On place ensuite... en  $i$  échanges ; misère !

## Tri insertion : recherche dichotomique

- On trouve la position de  $T[i]$  en  $\ln_2 i$  comparaisons.
- On place ensuite... en  $i$  échanges ; misère !
- Complexité améliorée en comparaisons, mais pas en temps.

# Tri insertion : recherche dichotomique

- On trouve la position de  $T[i]$  en  $\ln_2 i$  comparaisons.
- On place ensuite... en  $i$  échanges ; misère !
- Complexité améliorée en comparaisons, mais pas en temps.
- Borne théorique (sur les comparaisons) atteinte !

# Tri insertion : recherche dichotomique

- On trouve la position de  $T[i]$  en  $\ln_2 i$  comparaisons.
- On place ensuite... en  $i$  échanges ; misère !
- Complexité améliorée en comparaisons, mais pas en temps.
- Borne théorique (sur les comparaisons) atteinte !
- De fait : meilleure constante multiplicative devant le  $n^2$ .

# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- Tri bulle
- Sélection
- Insertion

## 3 Divide and conquer : deux tris récursifs

- **Tri fusion**
- Tri rapide

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----



# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

--	--	--	--	--	--

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

7					
---	--	--	--	--	--

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

7	10				
---	----	--	--	--	--

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

7	10	15			
---	----	----	--	--	--

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

7	10	15	20		
---	----	----	----	--	--

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

7	10	15	20	30	
---	----	----	----	----	--

# Tri fusion/dichotomique

10	40	7	30	20	15
----	----	---	----	----	----

COUIC

10	40	7
----	----	---

30	20	15
----	----	----

Magie de la récursion

7	10	40
---	----	----

15	20	30
----	----	----

Fusion

7	10	15	20	30	40
---	----	----	----	----	----



# Détails de la fusion

**Entrées :**  $T_1, T_2$

$T \leftarrow []$  ( $T$  va recevoir les listes fusionnées)

$(i_1, i_2) \leftarrow (0, 0)$  (les indices des cases à traiter)

**tant que**  $i_1 < |T_1|$  *ou*  $i_2 < |T_2|$  (*la fusion n'est pas finie*) **faire**

**si**  $i_2 \geq |T_2|$  *ou*  $(i_1 < |T_1|, i_2 < |T_1| \text{ et } T_1[i_1] < T_2[i_2])$  **alors**

        Adjoindre  $T_1[i_1]$  à  $T$

$i_1 \leftarrow i_1 + 1$

**sinon**

        Adjoindre  $T_2[i_2]$  à  $T$

$i_2 \leftarrow i_2 + 1$

# Tri fusion : complexité

- Séparation :  $O(1)$ ... ou  $O(n)$ .

# Tri fusion : complexité

- Séparation :  $O(1)$ ... ou  $O(n)$ .
- Deux appels récursifs.

# Tri fusion : complexité

- Séparation :  $O(1)$ ... ou  $O(n)$ .
- Deux appels récursifs..
- Fusion :  $O(n)$ .

# Tri fusion : complexité

- Séparation :  $O(1)$ ... ou  $O(n)$ .
- Deux appels récursifs..
- Fusion :  $O(n)$ .
- Pour simplifier :  $T(2n) = 2T(n) + n$   
(en fait,  $T(n) \leq M(n)$ , avec  $M(n) = M(\lceil n/2 \rceil) + M(\lfloor n/2 \rfloor)$ ...)

# Tri fusion : complexité

- Séparation :  $O(1)$ ... ou  $O(n)$ .
- Deux appels récursifs..
- Fusion :  $O(n)$ .
- Pour simplifier :  $T(2n) = 2T(n) + n$   
(en fait,  $T(n) \leq M(n)$ , avec  $M(n) = M(\lceil n/2 \rceil) + M(\lfloor n/2 \rfloor)$ ...)
- Considérer  $u_k = \frac{T(2^k)}{2^k} \dots$

# Tri fusion : complexité

- Séparation :  $O(1)$ ... ou  $O(n)$ .
- Deux appels récursifs..
- Fusion :  $O(n)$ .
- Pour simplifier :  $T(2n) = 2T(n) + n$   
(en fait,  $T(n) \leq M(n)$ , avec  $M(n) = M(\lceil n/2 \rceil) + M(\lfloor n/2 \rfloor)$ ...)
- Considérer  $u_k = \frac{T(2^k)}{2^k} \dots$
- $T(2^k) = O(k2^k)$ , puis  $T(n) = O(n \ln n)$ .

# Sommaire

## 1 Un peu de contexte

- De tout temps...
- Ce qui/qu'on compte
- Tour d'horizon

## 2 Tris quadratique

- Tri bulle
- Sélection
- Insertion

## 3 Divide and conquer : deux tris récursifs

- Tri fusion
- **Tri rapide**



# Tri rapide

- On va trier (récursivement et en place) une zone  $T[d : f + 1]$ .

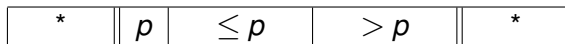


# Tri rapide

- On va trier (récursivement et en place) une zone  $T[d : f + 1]$ .



- On commence par casser cette zone en deux :

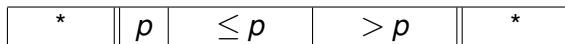


# Tri rapide

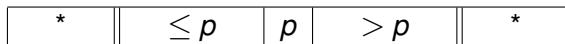
- On va trier (récursivement et en place) une zone  $T[d : f + 1]$ .



- On commence par casser cette zone en deux :



puis

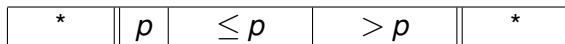


# Tri rapide

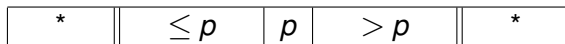
- On va trier (récursivement et en place) une zone  $T[d : f + 1]$ .



- On commence par casser cette zone en deux :



puis



- On réalise deux appels récursifs.

# Préparation de la zone

- LE (un) bon invariant :

*	$\rho$	$\leq \rho$	$T[i_1]$	$> \rho$	$T[i_2]$	*	*
---	--------	-------------	----------	----------	----------	---	---

# Préparation de la zone

- LE (un) bon invariant :

*	$p$	$\leq p$	$T[i_1]$	$> p$	$T[i_2]$	*	*
---	-----	----------	----------	-------	----------	---	---

- Comment initialiser  $i_1$  et  $i_2$  ?
- L'algorithme :

**Entrées :**  $T, d, f$

$i_1, i_2 \leftarrow d, d$  # Si, vraiment !

**tant que**  $i_2 < f$  **faire**

**si**  $T[i_2 + 1] \leq p$  **alors**

$T[i_1 + 1] \leftrightarrow T[i_2 + 1]$

$i_1 \leftarrow i_1 + 1$

$i_2 \leftarrow i_2 + 1$

$T[d] \leftrightarrow T[i_1]$

**Résultat :**  $i_1$

# Tri rapide : complexité

- $O(n)$  pour la préparation.

# Tri rapide : complexité

- $O(n)$  pour la préparation.
- $O(n^2)$  pour l'exécution globale « si ça se passe mal ».



## Tri rapide : complexité

- $O(n)$  pour la préparation.
- $O(n^2)$  pour l'exécution globale « si ça se passe mal ».
- On peut espérer  $O(n \ln n)$ ...
- Et « en moyenne », c'est effectivement le cas.

# C'est fini !

Merci de votre attention.

