ALG4 : algorithmes de tri

loig.jezequel@univ-nantes.fr

Context du cours

Problème général

Étant donnée une liste d'éléments, on souhaite trier ces éléments selon un ordre donné.

Par exemple

Trier des mots par ordre alphabétiques, des nombres du plus petit au plus grand, des couleurs de la plus foncée à la plus claire, etc.

Problème simplifié pour ce cours

Étant donné un tableau d'entiers t on souhaite produire un tableau t' qui contient les mêmes entiers rangés du plus petit au plus grand.

Par exemple

si $t =$ | 12 | 32 | 7 | 23 | 9 | alors $t' =$ | 7 | 9 | 12 | 23 | 32 |

Solution : algorithmes de tri

Tri par insertion

Mettre les éléments un par un, à la bonne place, dans un tableau initialement vide.

Tri fusion

Trier plusieurs sous ensembles d'éléments, puis les réunir en un seul tableau trié.

Tri rapide

Permutations d'éléments pour mettre chaque élément à la bonne place (tous les plus petits à sa gauche, tous les plus grands à sa droite).

Et plein d'autres

Tri à bulles, tri par tas, tri par sélection, etc.

Pourquoi plusieurs algorithmes différents ?

Tri en place ou non

Un tri est dit en place s'il est fait directement dans la structure de donnée à trier : il n'y a pas de recopie des données.

Tri stable ou non

Un tri est dit stable s'il n'altère pas l'ordre des éléments de même valeur. Ceci n'a pas d'importance sur un tableau d'entiers, mais peut en avoir si on trie des objets plus complexes (par exemple des étudiants qu'on trie par leur nom de famille).

Complexité dans le pire cas variable

Nombre d'opérations nécessaires dans le pire cas pour réaliser le tri.

Complexité en moyenne variable

Nombre d'opérations nécessaires en moyenne pour réaliser un tri.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.

12	32	7	23	9

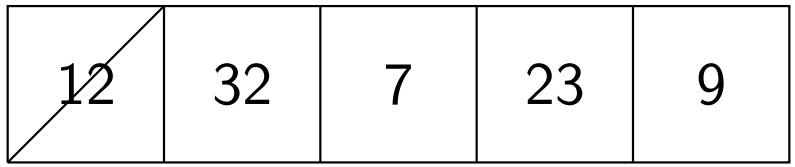
Algorithmes mis en œuvre

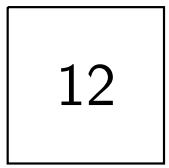
Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.





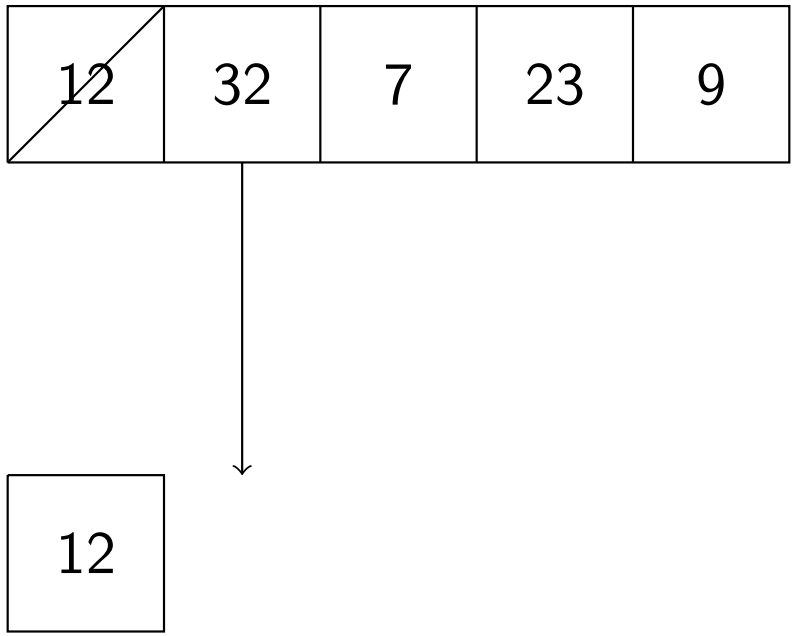
Algorithmes mis en œuvre

Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.



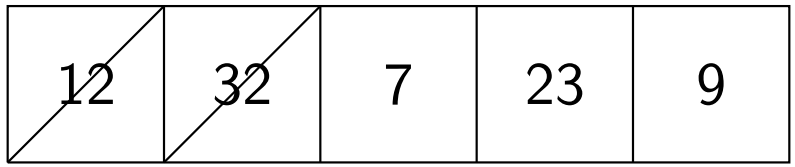
Algorithmes mis en œuvre

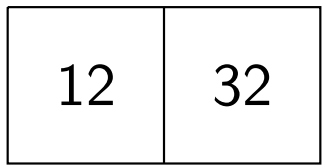
Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.





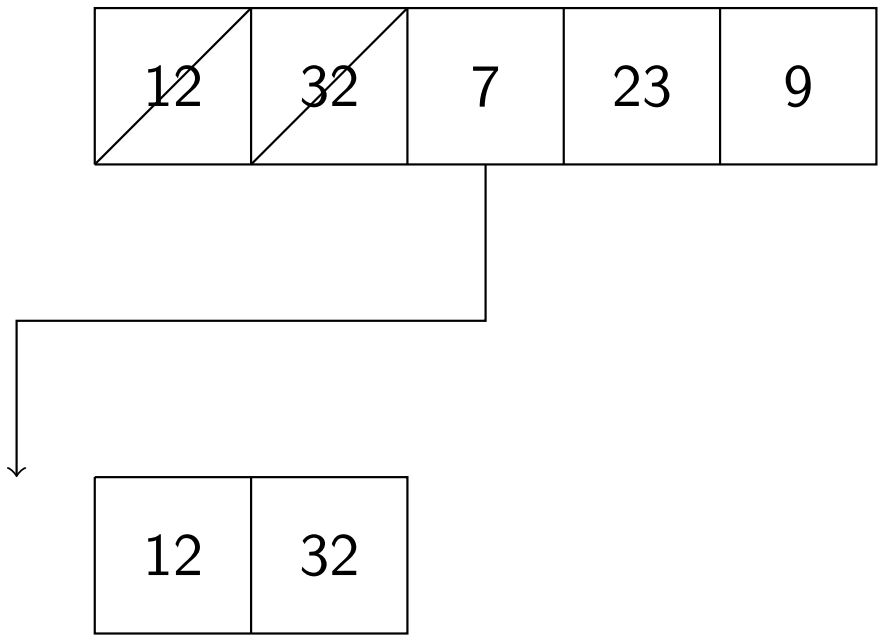
Algorithmes mis en œuvre

Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.



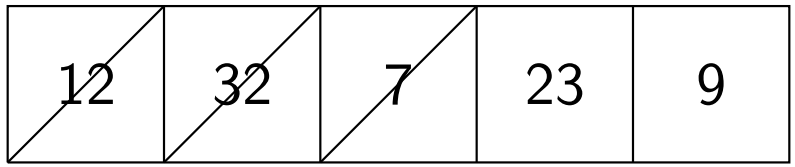
Algorithmes mis en œuvre

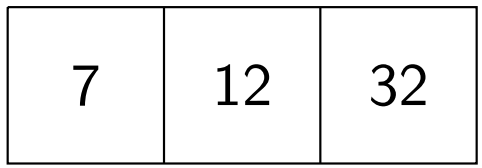
Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.





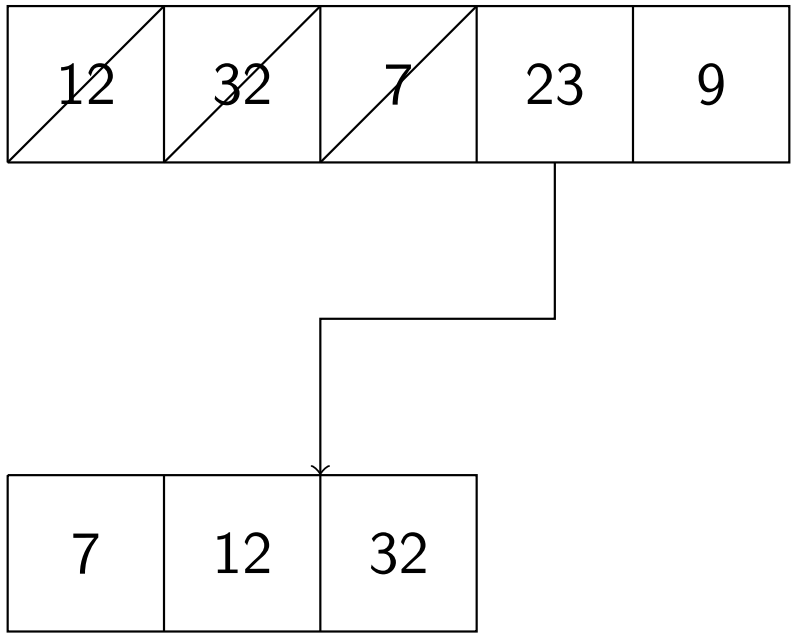
Algorithmes mis en œuvre

Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.



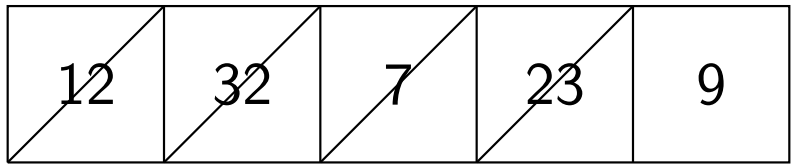
Algorithmes mis en œuvre

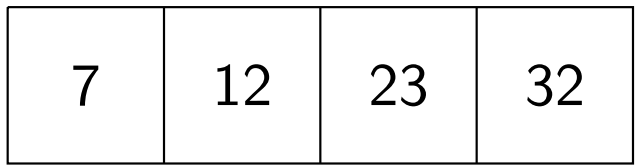
Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.





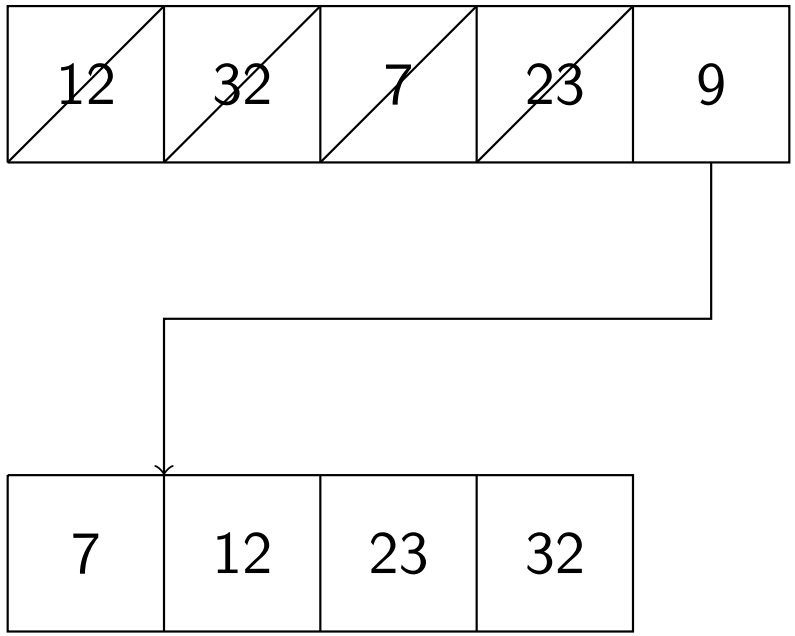
Algorithmes mis en œuvre

Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.



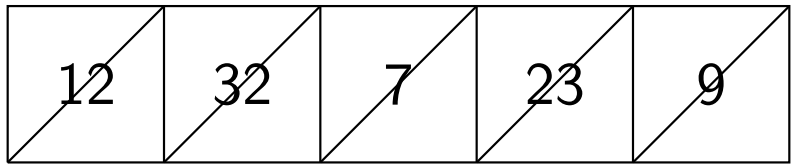
Algorithmes mis en œuvre

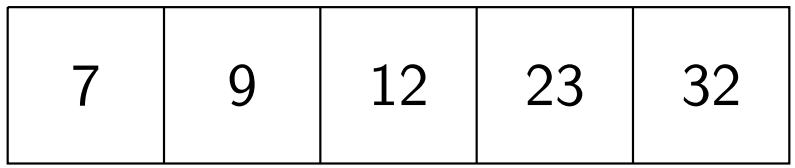
Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : principe

Idée générale

Construction d'un tableau trié un y ajoutant les entiers un à un, à la bonne place.





Algorithmes mis en œuvre

Parcours du tableau d'origine et recherche dans le tableau trié.

Le tri par insertion : algorithme

Entrées

Un tableau t d'entiers.

Sorties

Un tableau t' contenant les mêmes entiers que t rangés du plus petit au plus grand.

Tri par insertion

Soit t' un tableau vide. Soit $i = 0$. Tant que $i < len(t)$, ajouter $t[i]$ dans t', augmenter i de 1. Retourner t'.

Ajout de v dans t' trié

Soit $j = 0$.
Tant que $j < len(t')$ et $t'[j] \leq v$, augmenter j de 1.
Tant que $j < len(t')$, poser $tmp = t'[j]$, puis $t'[j] = v$ et enfin $v = tmp$, augmenter j de 1.
Ajouter v à la fin de t'.

Le tri par insertion : exemple

Tri de $t =$ | 12 | 32 | 7 | 23 | 9 |

- $i = 0$, ajout de $t[0] = 12$ à t'
  - 12 est placé à la fin de t'
  - $t' =$ | 12 |
- $i = 1$, ajout de $t[1] = 32$ à t'
  - $j = 0$, $t'[0] = 12 < 32$
  - 32 est placé à la fin de t'
  - $t' =$ | 12 | 32 |
- $i = 2$, ajout de $t[2] = 7$ à t'
  - $j = 0$, $t'[0] = 12 > 7$
  - $j = 0$, $t'[0] = 7$, $v = 12$
  - $j = 1$, $t'[1] = 12$, $v = 32$
  - 32 est placé à la fin de t'
  - $t' =$ | 7 | 12 | 32 |
- $i = 3$, ajout de $t[3] = 23$ à t'
  - $j = 0$, $t'[0] = 7 < 23$
  - $j = 1$, $t'[1] = 12 < 23$
  - $j = 2$, $t'[2] = 32 > 23$
  - $j = 2$, $t'[2] = 23$, $v = 32$
  - 32 est placé à la fin de t'
  - $t' =$ | 7 | 12 | 23 | 32 |
- $i = 4$, ajout de $t[4] = 9$ à t'
  - $j = 0$, $t'[0] = 7 < 9$
  - $j = 1$, $t'[1] = 12 > 9$
  - $j = 1$, $t'[1] = 9$, $v = 12$
  - $j = 2$, $t'[2] = 12$, $v = 23$
  - $j = 3$, $t'[3] = 23$, $v = 32$
  - 32 est placé à la fin de t'
  - $t' =$ | 7 | 9 | 12 | 23 | 32 |

Le tri par insertion : caractéristiques

Stable

Ce tri est stable (attention, si on change le $\leq$ dans l'insertion d'un élément dans un tableau trié par un $<$ le tri fonctionne toujours mais n'est plus stable).

Pas en place

Cette version du tri par insertion n'est pas en place, il est cependant possible de faire ce tri en place (on le voit juste après).

Opérations dans le pire cas

De l'ordre de $len(t)^2$.

Opérations en moyenne

De l'ordre de $len(t)^2$.

Le tri par insertion en place : principe

Lien avec l'algorithme précédent

t et t' sont stockés dans le même tableau : t' occupe l'espace libéré par les éléments de t qui ont déjà été triés.

Trier des cartes

C'est le tri qu'on utilise en général pour trier une main de cartes.

Un exemple en dansant

`https://www.youtube.com/watch?v=ROalU379l3U`

Le tri par insertion en place : algorithme

Entrées

Un tableau t d'entiers.

Sorties

le même tableau t contenant les mêmes entiers rangés du plus petit au plus grand.

Algorithme

Soit $i = 1$.
Tant que $i < len(t)$,
    soit $v = t[i]$,
    soit $j = i - 1$,
    tant que $j >= 0$ et $t[j] > v$,
        poser $t[j + 1] = t[j]$,
        diminuer j de 1;
    poser$t[j + 1] = v$,
    augmenter i de 1.

Le tri fusion : principe, algorithme

Fusionner des tableaux triés (t_1 et t_2 fusionnés dans t)

C'est très simple à faire et peu coûteux en nombre d'opérations.

- Comparer le premier élément de t_1 avec celui de t_2,
- mettre le plus petit à la première place libre dans t,
- le supprimer de son tableau d'origine,
- recommencer jusqu'à vider t_1 ou t_2,
- copier le reste des éléments dans t.

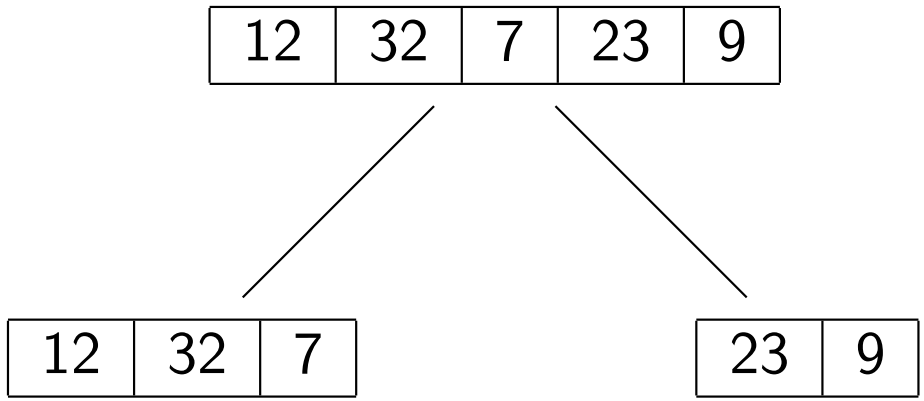
Tri fusion de t

- Séparer t en deux parties égales t_1 et t_2,
- trier t_1 par tri fusion (ou autre),
- trier t_2 par tri fusion (ou autre),
- retourner la fusion de t_1 et t_2.

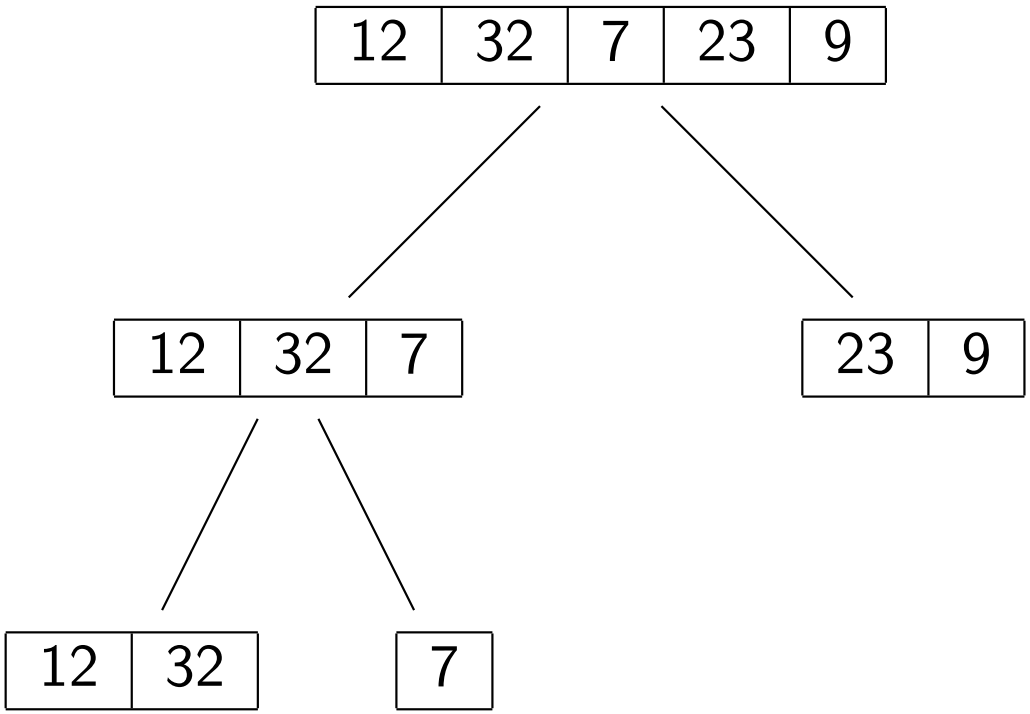
Le tri fusion : exemple

12	32	7	23	9

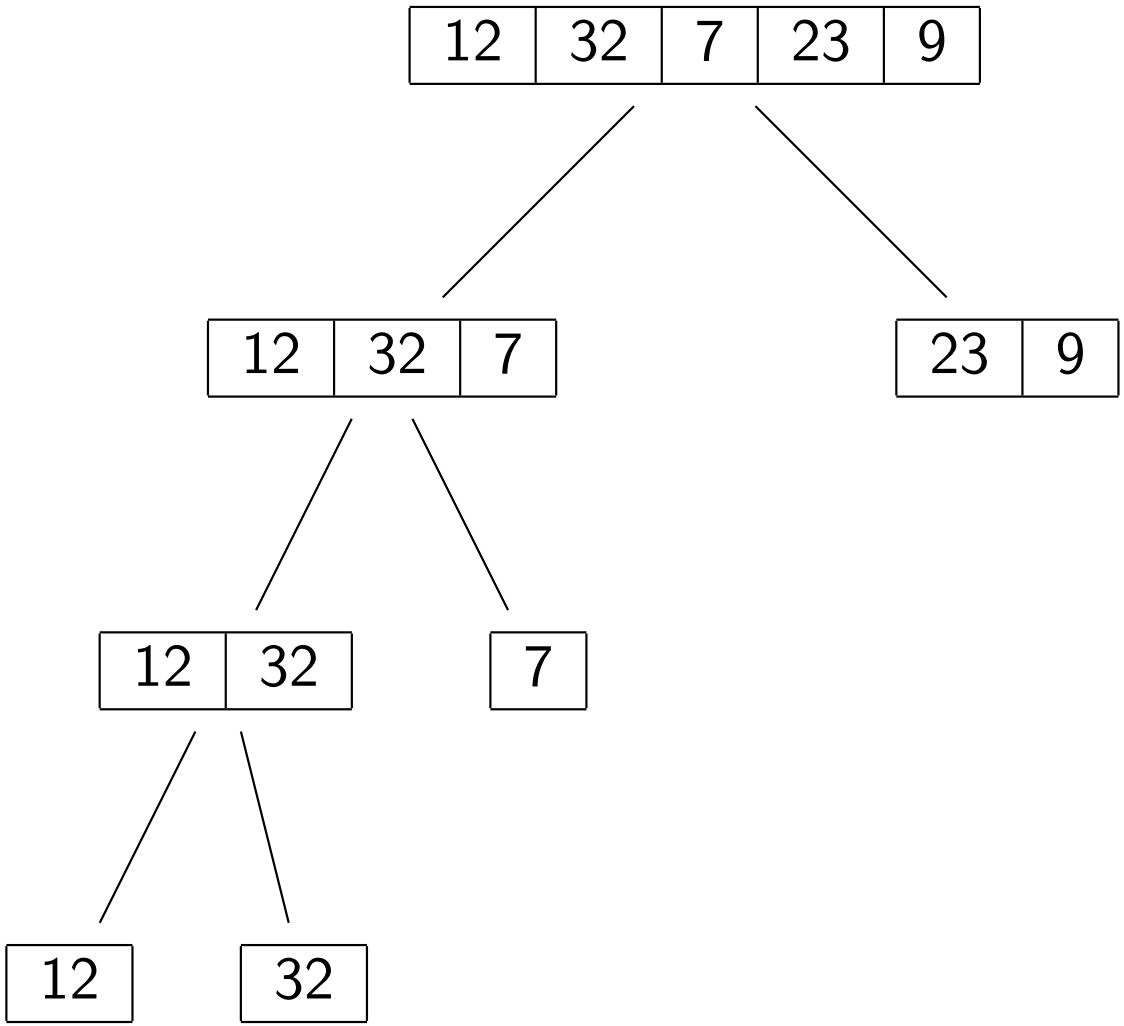
Le tri fusion : exemple



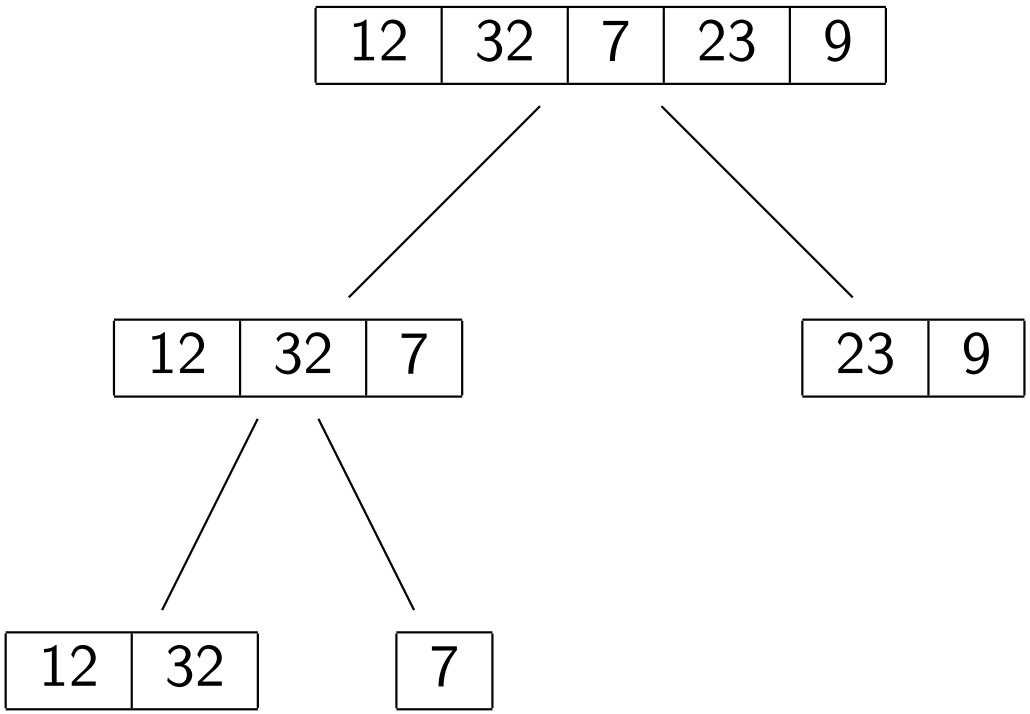
Le tri fusion : exemple



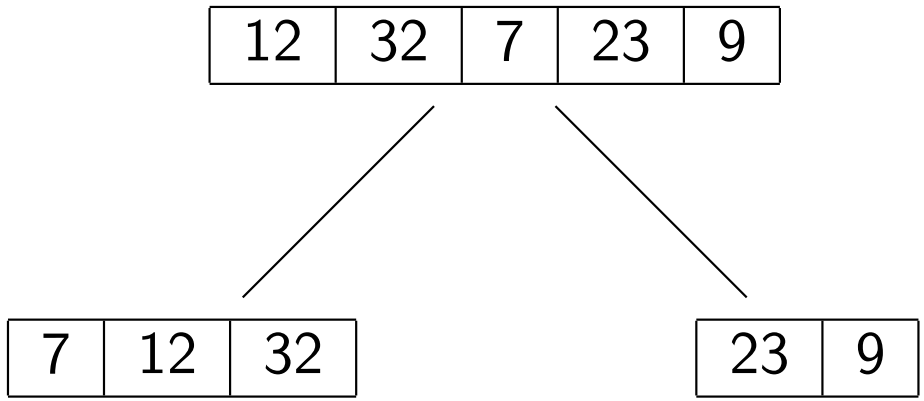
Le tri fusion : exemple



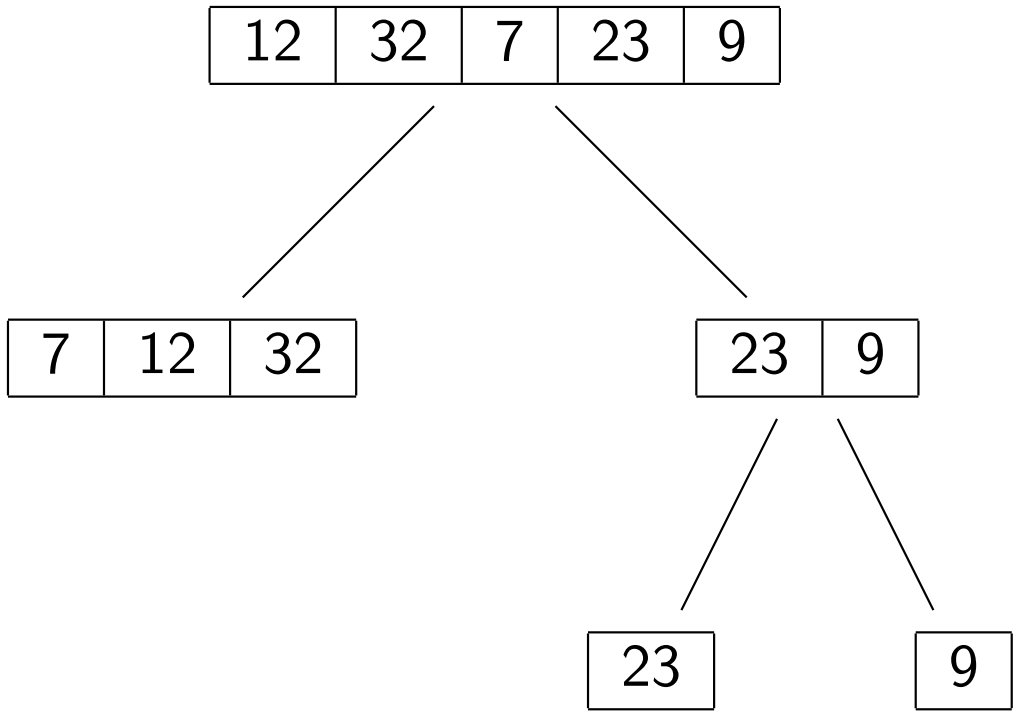
Le tri fusion : exemple



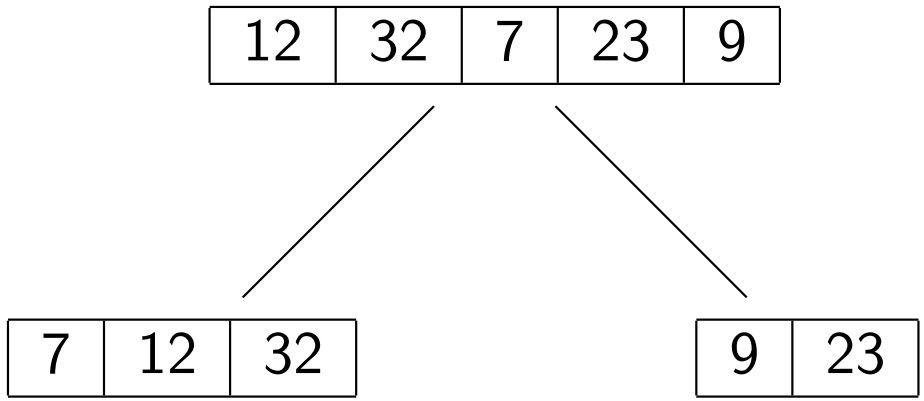
Le tri fusion : exemple



Le tri fusion : exemple



Le tri fusion : exemple



Le tri fusion : exemple

7	9	12	23	32

Le tri fusion : caractéristiques

Stable

Le tri fusion peut être implanté de façon à être stable.

En place

Avec un surcoût en nombre d'opérations dans le pire cas.

Opérations dans le pire cas

De l'ordre de $len(t)\log(len(t))$.

Opérations en moyenne

De l'ordre de $len(t)\log(len(t))$.

Approche diviser pour régner

Séparation d'un problème en sous-problèmes plus petits, combinaison des solutions à ces problèmes.

Le tri rapide : principe

Approche diviser pour régner

- Choisir un élément p de t, qu'on appelle pivot,
- Diviser le tableau t à trier en deux tableaux $t1$ et $t2$ tels que tous les éléments de $t1$ sont plus petits que p et tous les éléments de $t2$ sont plus grands que p,
- Trier $t1$ et $t2$ (éventuellement de la même façon),
- Mettre $t1$ et $t2$ bout à bout pour obtenir une version triée du tableau t.

En pratique ceci se fait en place, $t1$ et $t2$ sont stockés dans t, de part et d'autre de p.

Un exemple en dansant

`https://www.youtube.com/watch?v=ywWBy6J5gz8`

Le tri rapide : algorithme

Tri rapide de $t[p : r]$

Si $len(t[p : r]) > 1$, alors $q = partitionner(t[p : r])$, puis faire le tri rapide de $t[p : q]$ et de $t[q + 1 : r]$.

Partitionner($t[p : r]$)

Soit $pivot = t[p]$, soit $i = p$, soit $j = r - 1$.
Tant que $i < j$ faire,
    tant que $t[j] \geq pivot$ et $i < j$, diminuer j de 1;
    si $i < j$, inverser $t[i]$ et $t[j]$;
    tant que $t[i] \leq pivot$ et $i < j$, augmenter i de 1;
    si $i < j$, inverser $t[i]$ et $t[j]$.
Retourner i.

Le tri rapide : exemple

Tri rapide $t[0:5]$

12	32	7	23	9

Le tri rapide : exemple

Tri rapide $t[0:5]$

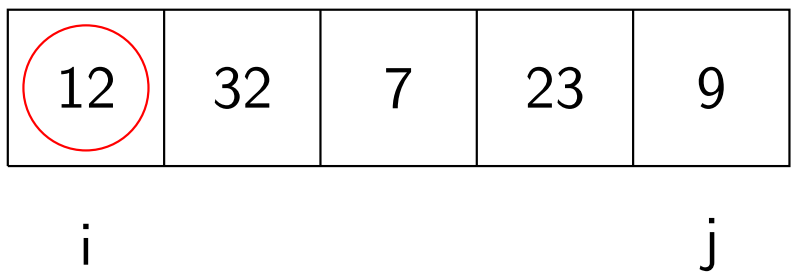
Partitionner($t[0:5]$)

12	32	7	23	9

Le tri rapide : exemple

Tri rapide $t[0:5]$

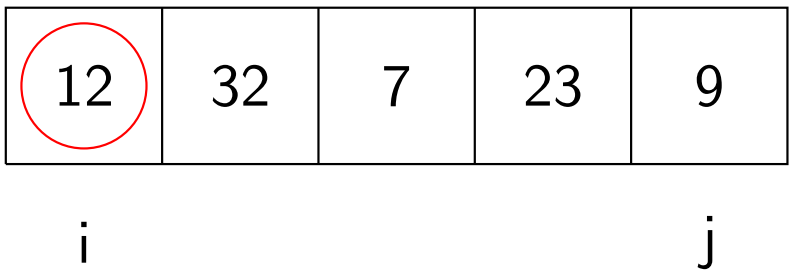
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

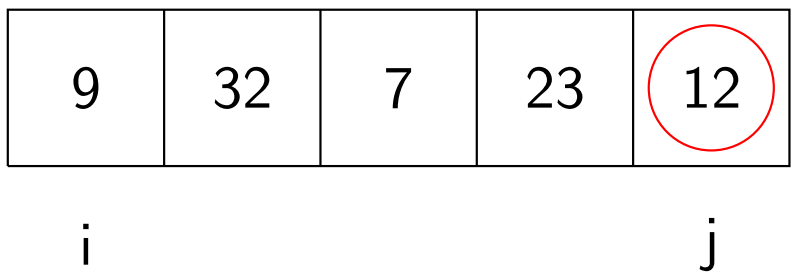
Partitionner($t[0:5]$) $9 < pivot$



Le tri rapide : exemple

Tri rapide $t[0:5]$

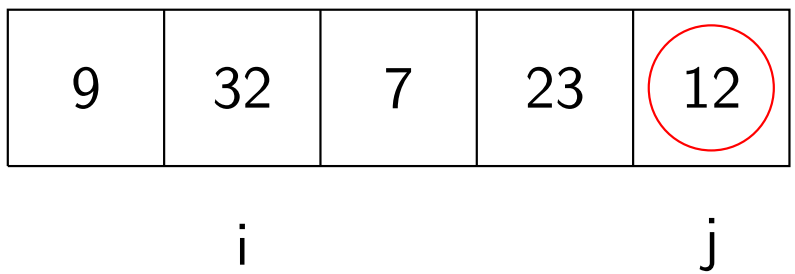
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

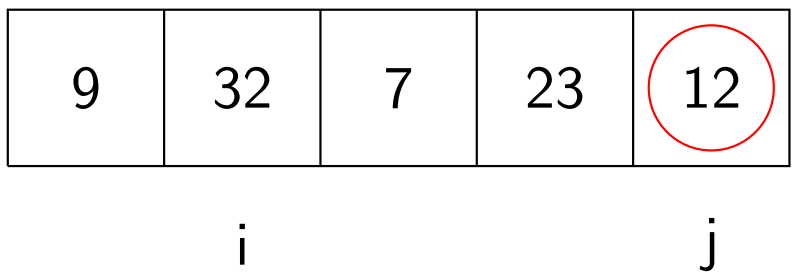
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

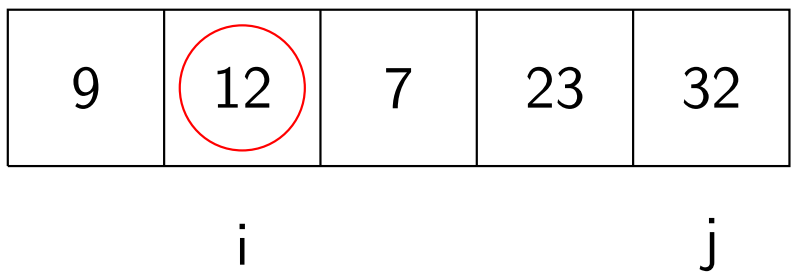
Partitionner($t[0:5]$) $32 > pivot$



Le tri rapide : exemple

Tri rapide $t[0:5]$

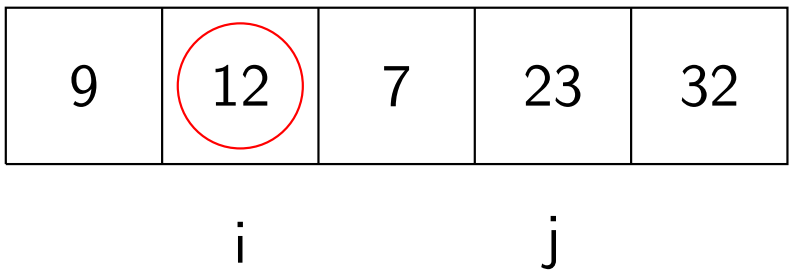
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

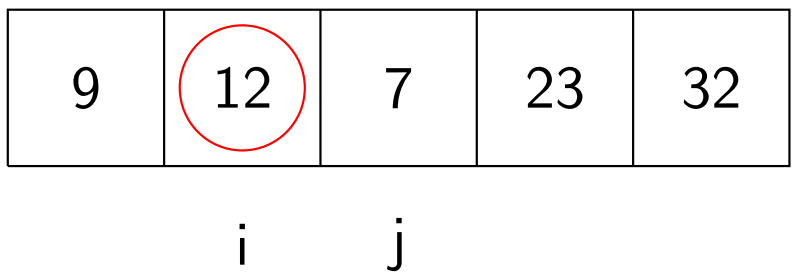
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

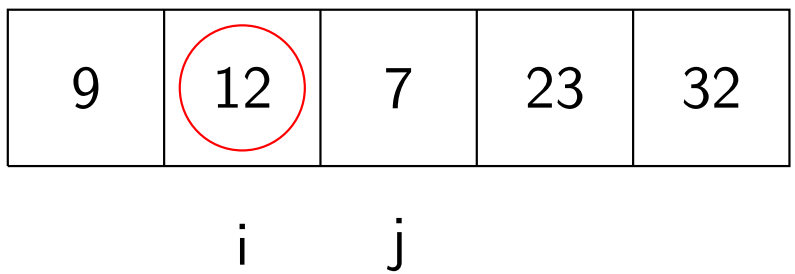
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

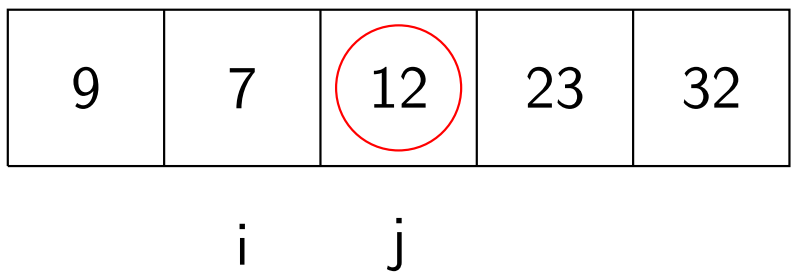
Partitionner($t[0:5]$) $7 < pivot$



Le tri rapide : exemple

Tri rapide $t[0:5]$

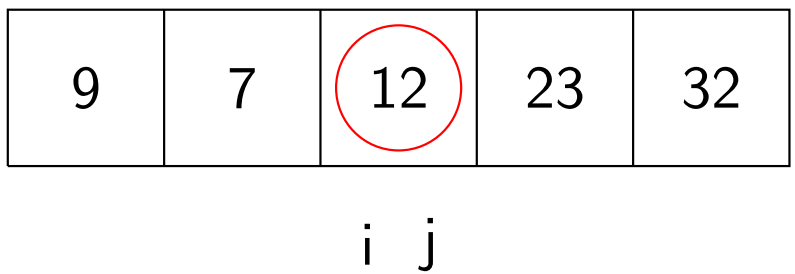
Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

Partitionner($t[0:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Tri rapide $t[3:5]$

9	7	12	23	32

Le tri rapide : exemple

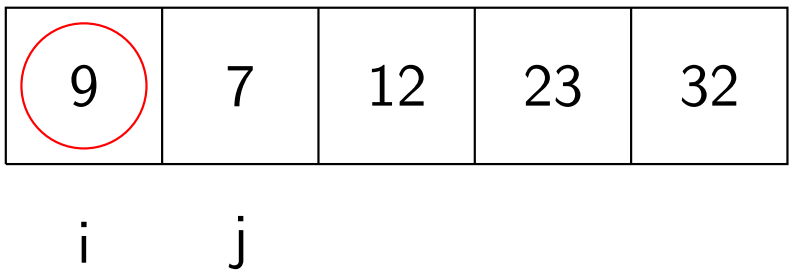
Tri rapide $t[0:5]$

Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$)

Tri rapide $t[3:5]$



Le tri rapide : exemple

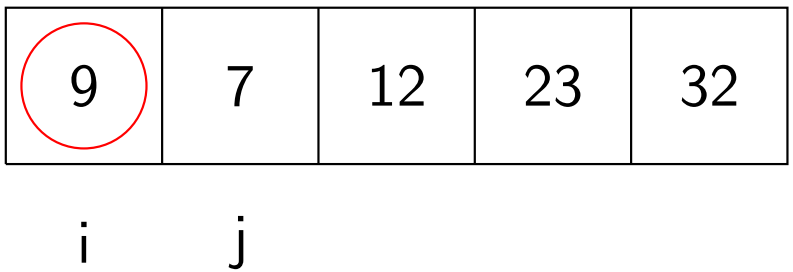
Tri rapide $t[0:5]$

Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$) $7 < pivot$

Tri rapide $t[3:5]$



Le tri rapide : exemple

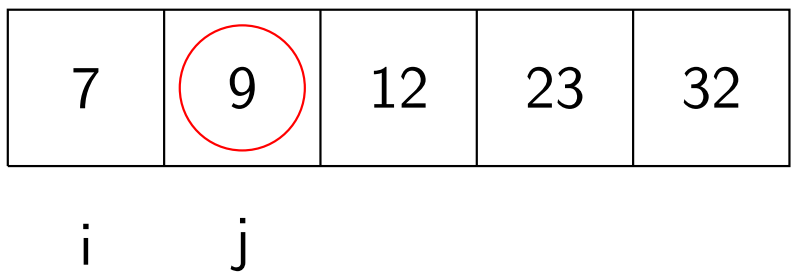
Tri rapide $t[0:5]$

Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$)

Tri rapide $t[3:5]$



Le tri rapide : exemple

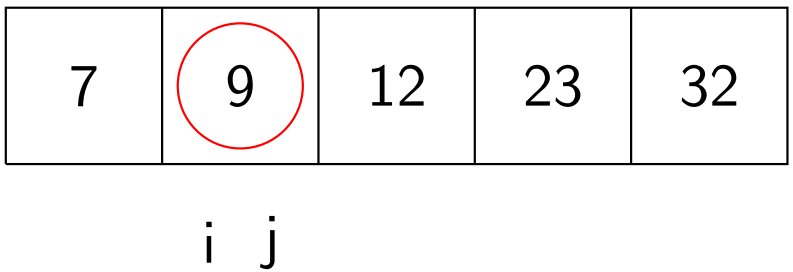
Tri rapide $t[0:5]$

Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$)

Tri rapide $t[3:5]$



Le tri rapide : exemple

Tri rapide $t[0:5]$

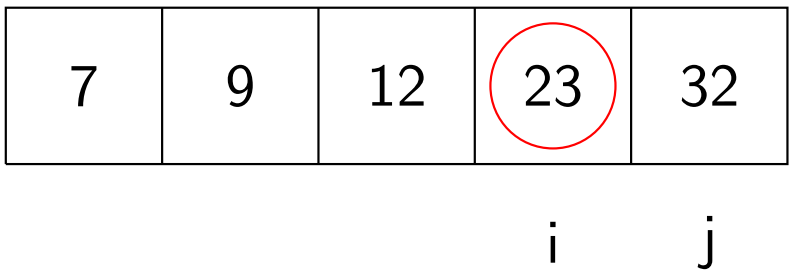
Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$)

Tri rapide $t[3:5]$

Partitionner($t[3:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

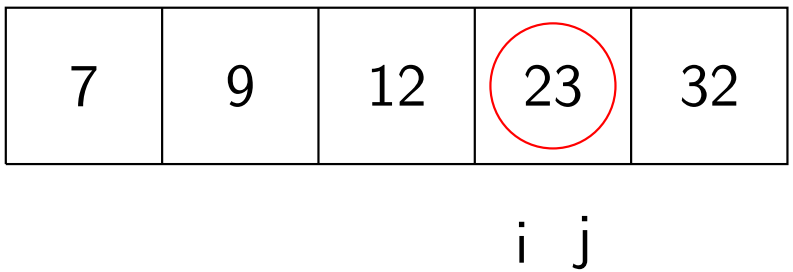
Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$)

Tri rapide $t[3:5]$

Partitionner($t[3:5]$)



Le tri rapide : exemple

Tri rapide $t[0:5]$

Partitionner($t[0:5]$)

Tri rapide $t[0:2]$

Partitionner($t[0:2]$)

Tri rapide $t[3:5]$

Partitionner($t[3:5]$)

7	9	12	23	32

Le tri rapide : caractéristiques

Pas stable

Le tri rapide n'est pas stable.

En place

Le tri rapide est effectué en place.

Opérations dans le pire cas

De l'ordre de $len(t)^2$.

Opérations en moyenne

De l'ordre de $len(t)\log(len(t))$.

Efficace en pratique

En pratique le pire cas est rarement atteint et les coefficients du nombre d'opérations en moyenne sont petits par rapport aux autres tris, ce qui fait qu'on préfère souvent le tri rapide.