

Algorithme des triés

Tri par insertion: (itératif)

Procédure insertion (T[n])

Pour i allant de 0 à n-1 faire

$x \leftarrow T[i]$

$j \leftarrow i - 1$

Tant que $(j > 0)$ et $(T[j] > x)$ faire

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

$j \leftarrow j - 1$

Fin tant que

$T[j+1] \leftarrow x$

Fin Pour

Tri par insertion (récursif)

Tri-Insertion (T: tableau, n: entier)

Debut

Si $(n > 1)$ alors

Tri par sélection (itératif)

Procédure sélection (T[n])

Pour i allant de 0 à n-1 faire

petit $\leftarrow i$

Pour j allant de i+1 à n-1 faire

si $T[j] < T[\text{petit}]$ alors

petit $\leftarrow j$

Fin si

Fin Pour

temp $\leftarrow T[\text{petit}]$

$T[\text{petit}] \leftarrow T[i]$

$T[i] \leftarrow \text{temp}$

Fin Pour

Tri par sélection (récursif)

Tri-Selection (T: tableau, debut, fin: entiers)

Debut

Si $(\text{debut} < \text{fin})$ alors

Tri par propagation (bubbles) (itératif)

Procédure tri-Bulle (T[n])

Pour i allant de n-2 à 0 faire

Pour k allant de 0 à i

Si $(T[k] > T[k+1])$ alors

temp $\leftarrow T[k]$

$T[k] \leftarrow T[k+1]$

$T[k+1] \leftarrow \text{temp}$

Fin si

Fin Pour

Tri à Bulles (récursif)

Tri-Bulle (T: tableau, n: entier)

Debut

Pour i de 0 à n

Si $T[i] > T[i+1]$ alors

permuter $(T[i], T[i+1])$

Fin si

Fin Pour

Tri-Insertion (T: tableau, entier)

Debut

Si $(n > 1)$ alors

Tri-Insertion(T, n-1);

Insère(T, n) // insère l'élément est à la place de la tab T[0...n-1]

Fin Si

Fin

Calcul de Complexité:

Meilleur Cas Moyen des cas Pire des cas
↓ ↓ ↓
(Boucle Pour Selu) (Pour + tout pas) (H l'alge)

$O(n)$

$$T_{\text{worst}}(n) = T(n-1) + \sum_{i=1}^{n-1} i$$

$O(n^2)$

Tri par sélection (Vocabulaire)

Tri Selection (T: tableau, debut, fin: entier)

Debut

Si (debut < fin) alors

petit ← Recherche-petit(T, debut, fin)

Permuter(T, debut, petit)

Tri-Selection(T, debut+1, fin)

Fin Si

Fin

Calcul de Complexité:

Meilleur Moyen Pire
↓ ↓ ↓
(H l'algo (Pour O Pour))

$O(n^2)$

Se (La) > (Lent) alors
permute(T[i], T[i+1])

Tri-Bulle (T, n-1)

Fin Si

Fin

Calcul de complexité

Meilleur Moyen Pire
↓ ↓ ↓
 $O(n)$ $O(n^2)$

$$T_{\text{worst}}(n) = \sum_{i=1}^{n-1} i$$

Algorithme de Tri :

Tri partition fusion (itératif)

Procédure Tri-fusion (T, deb, fin)

Début
i ← deb; j ← milieu + 1; k ← deb

Tant que (k ≤ fin) faire

Si (i ≤ milieu) alors

Si (j ≤ fin) alors

Si (T[i] < T[j]) alors

Temp[k] ← T[i]

i ← i + 1;

sinon

Temp[k] ← T[j]

j ← j + 1;

Fin si

sinon

Temp[k] ← T[i]

i ← i + 1;

Fin si

sinon

Temp[k] ← T[j]

Recherche dichotomique (itérative)

Fonction RechercheDicho(T, indice inf, indice sup, X, entier)

Variables :

Trouvé : Boolean; milieu : entier;

Début :

Trouvé ← false;

Tant que ((trouvé = false) ET (indice inf ≤ indice sup)) faire

milieu ← (indice inf + indice sup) / 2;

D.V. 2;

Si (Tab[milieu] = X) alors

trouvé ← true;

Sinon si (X < Tab[milieu]) alors

indice sup ← milieu - 1;

Sinon

indice inf ← milieu + 1;

Fin si

Fin si

Si (trouvé) alors

Retourner (milieu)

Complexité Pire cas

cas 1
K itération = $\frac{n}{2^{K-1}}$

$T_n(n) = C_1 K + C_2$

Tri rapide de l'entier

Fonction tri rapide (T, deb, fin)

Début :

eltPivot ← T[deb];

i ← deb, j ← fin;

Tant que (i < j)

Tant que (T[i] < eltPivot)

i ← i + 1;

Fin Tant que

Tant que (T[j] > eltPivot)

j ← j - 1;

Fin Tant que

Si (i < j) alors

Temp ← T[i]

T[i] ← T[j]

T[j] ← Temp;

sinon

retourner j;

Fin si

Fin

Temp(K) = T(k)
co = i + 1;

Fin si

Si non

replk la T(j)

Fin si

K = K + 1

Fin tant que

↓

Tri partition fusion (récurssive)

Procédure Tri-fusion (T, deb, fin)

Variables: n: entier

Début:

si (deb < fin) alors

milieu ← (deb + fin) DIV 2;

Tri-fusion (T, deb, milieu, T(n/2))

Tri-fusion (T, milieu + 1, fin, T(n/2))

Fusionner (T, deb, milieu, fin, T(n))

Fin si

Fin

avec Fusionner (T, deb, milieu, fin)

{*}

Pour l allant de début à fin faire

T(l) ← temp(l);

Fin Pour $\mathcal{O}(n)$

Fin

Fin si

Fin si

si (verifier) alors

Retourner (milieu)

Si non

Retourner (-1);

Fin si

Fin Recherche dich

Recherche dich. (récurssive)

Function Rech Dich (T, indice inf, indice sup, x, left)
: Boolean

si (indice inf = indice sup) alors

mil ← (indice inf + indice sup) DIV 2; indice-pivot ← Part. triant T, deb, fin)

si (Tab[mil] = elt) alors

retourner (vrai)

Si non

si (Tab[mil] > x) alors

Retourner (Rech Dich (T, indice inf, mil - 1, x))

Si non

Retourner (Rech Dich (T, mil + 1, indice sup, x))

Fin si

Si non

Retourner (Faux)

Fin si

T(i) = T(j)

T(j) = Temp

Si non

retourner j;

Fin si

Fin

Tri-rapide (récurssive)

Procédure Tri-rapide (T, deb, fin)

Variables

indice-pivot

Début:

si (deb < fin) alors

Part. triant T, deb, fin)

Tri-rapide (T, deb, indice-pivot - 1)

Tri-rapide (T, indice-pivot + 1, fin)

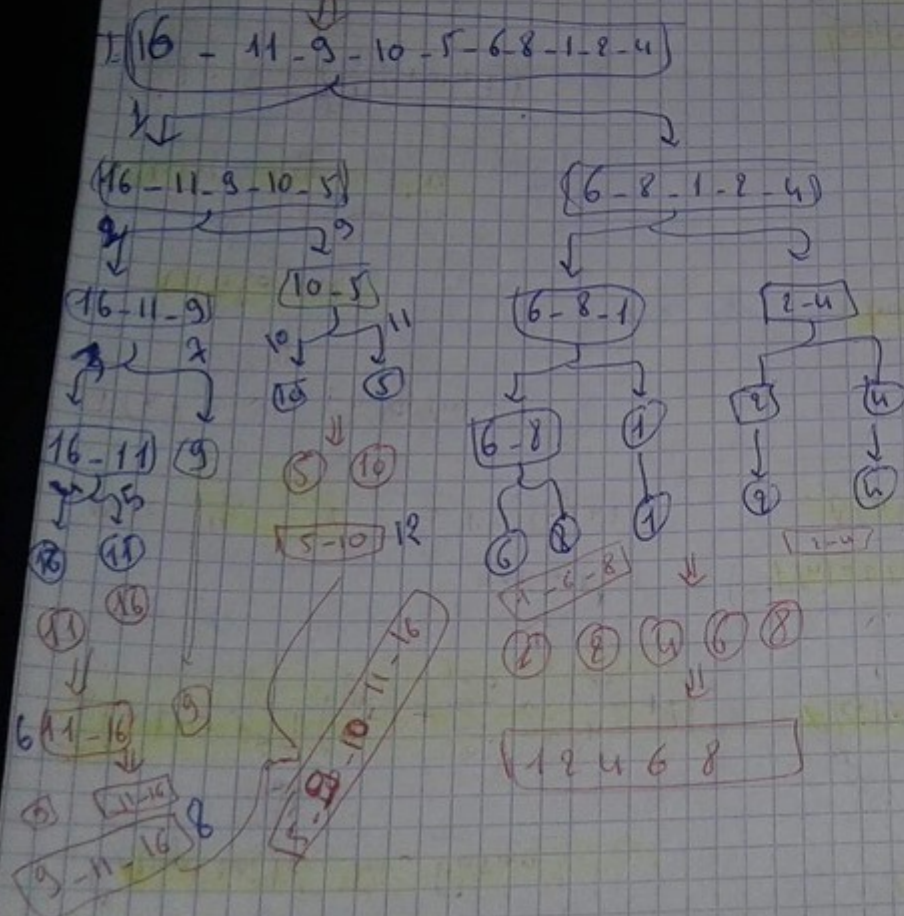
Fin si

Fin

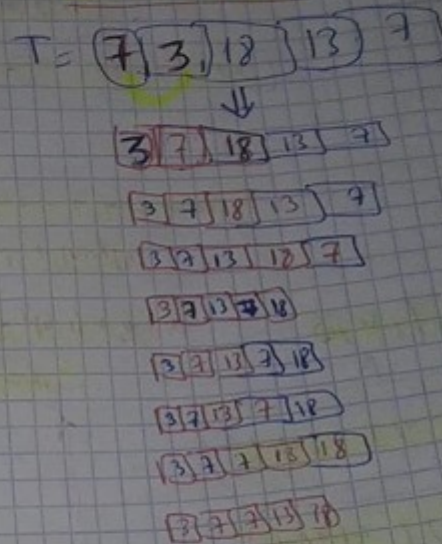
avec Part. triant → (*) $\mathcal{O}(n)$

01325

Die Funktion



Tr. a. Bulle



$$T_{\text{ex}}(n) = O(c)$$

$$\Rightarrow \exists C_1 > 0 \text{ et } n \gg n_0$$

$$T_{\text{exp}}(n) \leq G.C$$

Application is algo:

Trie par insertion

$T = [3, 1, 4, 0, 5]$

$x \leftarrow T[1] = 1$

3 3 4 0 5

1 3 4 0 5

$x \leftarrow T[2] = 4$

1 3 3 0 5

1 3 4 0 5

$x \leftarrow T[3] = 0$

1 3 4 4 5

1 3 3 4 5

1 1 3 4 5

0 1 3 4 5

$x \leftarrow T[4] = 5$

0 1 3 4 4

0 1 3 4 5

Trie par selection

$T = [8, 5, 2, 1, 0]$

petit $\leftarrow 0$ $8 > 5$
petit $\leftarrow 1$ $8 > 2$
petit $\leftarrow 2$ $8 > 1$
petit $\leftarrow 3$ $8 > 0$
petit $\leftarrow 4$

0 5 2 1 8

petit $\leftarrow 1$ $5 > 2$
petit $\leftarrow 2$ $5 > 1$
petit $\leftarrow 3$

0 1 2 5 8

petit $\leftarrow 2$ $(0) 2 < 5, 2 < 8$ non fait

0 1 2 5 8

petit $\leftarrow 3$ $5 < 8$

0 1 2 5 8

Trie rapide $i=0, j=3$

10 6 11 4

elb pivot = $T[0] = 10$
 $i=0, j=3$ $(10 > 4)$

4 6 11 10

$i=1, j=3$ $(6 < 10)$

$i=2, j=3$ $(11 > 10)$

4 6 10 11

$i=2, j=2$ To 3th place on array

$i=0, j=1$

4 6

$i=1, j=1$

4 6

$i=3$

11

4 6

11

4 6 10 11

Complexité Tri-Fusion :

$$T(n) = 2^k T\left(\frac{n}{2^k}\right) + \sum_{i=0}^{k-1} 2^i \left(\frac{n}{2^i} + 1\right)$$

$$= n \times k + 2^k - 1$$

avec $\frac{n}{2^k} = 1 \Rightarrow k = \log(n)$

$$T(n) = 2n + n \log(n) - 1$$

$$\Downarrow$$

$$O(n \log(n))$$

Meilleure Moyenne Pire

$$T(n) = 1 + 2\left(\frac{n}{2}\right) + n + 1$$

Complexité Tri-rapide

Meilleure Cas

$$T(n) = 2T\left(\frac{n}{2}\right) + n + 1$$

$$= O(n \log(n))$$

Moyenne

cas moyenne

$$T_{\text{moy}}(n) = \frac{1}{n} \left[T_{\text{moy}}(n-1) + T_{\text{moy}}(1) \right. \\ \left. + T_{\text{moy}}(n-2) + T_{\text{moy}}(2) \right. \\ \left. + T_{\text{moy}}(n-3) + \dots + T_{\text{moy}}(n-1) \right. \\ \left. + n + 1 \right]$$

$$T_{\text{moy}}(n) = (n+1) \left(-\frac{1}{2} + 2 \log(n) + c \right)$$

$$T_{\text{moy}}(n) = O(n \log(n))$$

Pire des cas

$$T(n) = T(1) + \sum_{i=1}^n 2i$$

$$T(n) = T(n-1) + n + 1$$

$$O(n^2)$$

$$T(n) = O(n \log n)$$

Fonction Element Central ($T_1, T_2, d_1, f_1, d_2, f_2$) entree / sortie (lignes)

Variables :

M , lien 1, M , lien 2 2 entrees

Debut :

$$\text{milieu 1} \leftarrow \frac{d_1 + f_1}{2}$$

$$\text{milieu 2} \leftarrow \frac{d_2 + f_2}{2}$$

Si $T_1[\text{milieu 1}] < T_2[\text{milieu 2}]$ alors

$$d_1 \leftarrow \text{milieu 1} + 1$$

$$f_2 \leftarrow \text{milieu 2}$$

sinon

$$d_2 \leftarrow \text{milieu 2} + 1$$

$$f_1 \leftarrow \text{milieu 1}$$

Fin Si

retour Element Central ($T_1, T_2, d_1, f_1, d_2, f_2$)

recommencer (retour)

Fin Fonction