

Les algorithmes de tri

Tri par sélection (ordre croissant)

Algorithme :

procédure trier (@T : tab, n : entier)

Début

Pour i de 0 à n-2 faire

 indice $\leftarrow$ i

 Pour j de i+1 à n-1 faire

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

 indice $\leftarrow$ j

 FinSi

FinPour

Si (Indice $\neq$ i) alors

 aux $\leftarrow$ T[i]

 T[i] $\leftarrow$ T[indice]

 T[indice] $\leftarrow$ aux

FinSi

FinPour

Fin

Python :

Trier le tableau T dans l'ordre croissant (tri par sélection)

def trier(T,n):

 for i in range(0,n-1):

 indice=i

 for j in range(i+1,n):

 if T[j] < T[indice]:

 indice=j

 if indice!=i:

 T[i], T[indice]=T[indice], T[i]

Déclaration des objets locales :

Objet	Type/nature
i	Entier
j	Entier
indice	Entier
aux	Entier

Tri à bulles (ordre décroissant)

Tri à bulles version1

Algorithme :

Procédure trier (@T : tab, n : entier)

Début

permut ← vrai

i ← 0

Tantque permut=vrai faire

permut ← faux

i ← i+1

Pour j de 0 à n-1 -i faire

Si (T[j] < T[j+1]) alors

aux ← T[j]

T[j] ← T[j+1]

T[j+1] ← aux

permut ← Vrai

FinSi

FinPour

FinTantque

Fin

TDOL

Objet	Type/nature
i	Entier
j	Entier
aux	Entier
permut	booléen

Python :

def trier(T, n):

permut = True

i = 0

while permut:

permut = False

i = i+1

for j in range(n-i):

if T[j] < T[j+1]:

T[j], T[j+1] = T[j+1], T[j]

permut = True

Tri à bulles version2

Algorithme :

Procédure trier (@T : tab, n : entier)

Début

pour i de 0 à n-2 faire

pour j de 0 à n-2-i faire

Si (T[j] < T[j+1]) alors

aux ← T[j]

T[j] ← T[j+1]

T[j+1] ← aux

FinSi

FinPour

FinPour

Fin

TDOL

Objet	Type/nature
i	Entier
j	Entier
aux	Entier

Python :

def trier(T, n):

for i in range(n-1):

for j in range(n-1-i):

if T[j] < T[j+1]:

T[j], T[j+1] = T[j+1], T[j]