Chapitre n°6 :

Les traitements avancés

Leçon 1 — Méthodes de tri

Introduction

Un algorithme de tri permet de mettre dans un ordre croissant ou décroissant des valeurs stockées dans une structure de donnée comme un tableau en mémoire centrale. Ces valeurs pourraient être de type numérique ou caractères. On recense une dizaine de méthodes de tri.

Simulation : http://lwh.free.fr/pages/algo/tri/tri.htm

I – Tri par sélection

Énoncé

Faire l'analyse d'un programme qui permet de trier un tableau T de n éléments dans **l'ordre croissant**. La valeur maximum de n est 50.

Principe

Sa consiste à trouver l'emplacement de l'élément le plus petit du tableau et le permuter avec le premier élément du tableau. Ce même procédé est réutilisé pour trier le reste du tableau.

a. Analyse principale

	Nom : tri_selection	
S	L.D.E	O.U
4	Résultat = PROC affichage (T, n)	T, n
3	T = PROC tri (T, n)	affichage
2	T = PROC remplir (T, n)	tri
1	n = PROC saisie (n)	remplir
5	Fin tri_selection	saisie

T.D.N.T

Type
Tab = tableau de 50 entiers

T.D.O globaux

Objet	Type	Rôle
n	Entier	Taille du tableau
T	Tab	Tableau d'entiers
affichage	Procédure	Affichage du tableau trié
Tri	Procédure	
Remplir	Procédure	Remplissage de T
Saisie	Procédure	Saisie contrôlée de n

Algorithme :

0) Début tri_selection
1) PROC saisie (n)
2) PROC remplir (T, n)
3) PROC tri (T, n)
4) PROC affichage (T, n)
5) Fin tri_selection

b. Analyse de la procédure tri – (Ordre croissant)

	DEF PROC tri (var T : tab, n : entier)	
S	L.D.E	O.U
1	Résultat = T T = [ ] pour i de 1 à n-1 faire pmin ← FN recherche_pmin (T, n, i) PROC permuter (T[i], T[pmin]) Fin pour i = compteur	i recherche_pmin pmin permuter
2	Fin tri	

T.D.O Locaux

Objet	Type	Rôle
i	Entier	compteur
Permuter	Procédure	
Recherche_pmin	Fonction	
pmin	Entier	Position du minimum

Algorithme :

0) DEF PROC tri (var T : tab, n : entier)
1) Pour i de 1 à n faire
   pmin ← FN recherche_pmin (T, n, i)
   PROC permuter (T[i], T[pmin])
   Fin pour
2) Fin tri

c. Analyse de la fonction recherche_pmin

	DEF FN recherche_pmin (T: tab, n: entier, i: entier): entier	
S	L.D.E	O.U
	Résultat = recherche_pmin	
2	recherche_pmin ← p	p
1	p = [p ← i] pour j de i+1 à n faire Si T[p] > T[j] alors p ← j Fin si Fin pour j = compteur	j
3	Fin recherche_pmin	

T.D.O Locaux

Objet	Type	Rôle
j	Entier	compteur
P	Entier	Position du minimum

Algorithme :

0) DEF FN recherche_pmin (T: tab, n: entier, i: entier): entier
1) p ← i
   Pour j de i+1 à n faire
   Si T[p] < T[j] alors p ← j
   Fin si
   Fin pour
2) recherche_pmin ← p
3) Fin recherche_pmin

d. Analyse de la procédure permuter

	DEF PROC permuter (var a : entier, var b : entier)	
S	**L.D.E**	**O.U**
1 2 3 4	Résultat = (a, b) p ← a a ← b b ← p Fin permuter	p

T.D.O Locaux

Objet	Type	Rôle
P	Entier	Variable tampon

Algorithme

0) DEF PROC permuter (var a : entier, var b : entier)
1) p ← a
2) a ← b
3) b ← p
4) Fin permuter

e. Analyse de la procédure affichage

	DEF PROC affichage (T : tab, n : entier)	
S	**L.D.E**	O.U
1 2	Résultat = Affichage Affichage = [ ] pour c de 1 à n faire Ecrire (T[c]) Fin pour c = compteur Fin affichage	c

T.D.O Locaux

Objet	Type	Rôle
c	Entier	Compteur

Algorithme

0) DEF PROC affichage (T : tab, n : entier)
1) pour c de 1 à n faire
   Ecrire (T[c])
   Fin pour
2) Fin affichage

f. Pascal (voir fichier : tri_sel.pas)

II – Tri à bulles

Énoncé :

Faire l'analyse d'un programme qui permet de trier un tableau T de n éléments dans **l'ordre croissant**. La valeur maximum de n est 50.

Principe :

Cet algorithme compare les éléments du tableau deux à deux, jusqu'à obtenir un tableau trié. Dans ce cas, la répétition de plusieurs passages sur l'ensemble des données est nécessaire pour l'obtention d'un tri complet. Cette méthode est appelée le tri à bulles, car les éléments mal classés remontent dans la liste comme des bulles à la surface d'un liquide.

a. Analyse de la procédure tri_bulles :

	DEF Proc tri_bulles (var T : tab, n : entier)	
S	**L.D.E**	O.U
1	Résultat = T T = [ ] Répéter [changer ← faux] pour i de 1 à n-1 faire Si T [i] > T [i+1] alors PROC permuter (T[i], T [i+1]) changer ← vrai Fin si Fin pour Jusqu'à (change = faux) i = compteur	changer i permuter
2	Fin calcul	

T.D.O Locaux

Objet	Type	Rôle
Changer	Booléen	Teste de l'état des permutations
i	Compteur	
permuter	procédure	

Algorithme

```
0) DEF PROC tri_bulles (var T : tab, n : entier)
1) Répéter
        changer ← faux
        pour i de 1 à n-1 faire
            Si T [i] > T [i+1] alors PROC permuter (T[i], T [i+1])
                                    changer ← vrai
            Fin si
        Fin pour
   Jusqu'à (change = faux)
2) Fin tri_bulles
```

b. Pascal (voir fichier : tri_bull.pas)

III – Tri par insertion

Énoncé :

Faire l'analyse d'un programme qui permet de trier un tableau T de n éléments dans **l'ordre croissant**. La valeur maximum de n est 50.

Principe :

p est la valeur à insérer dans l'endroit approprié du tableau à droite pour vider une place pour p. Finalement la valeur p est insérée à son emplacement adéquat.

a. Analyse de la procédure tri_insertion :

	DEF PROC tri_insertion (var T : tab, n : entier)	
S	**L.D.E**	O.U
1	Résultat = T T = [ ] pour i de 2 à n faire [p ← T [i], j ← i]	i
	Tant que (j – 1 ≥ 1) et (p < T[j – 1]) faire	p
	T [j] ← T [j – 1] j ← j – 1	j
	Fin Tant que	
	T [j] ← p	
	Fin pour	
	i = compteur	
2	Fin tri_insertion	

T.D.O Locaux

Objet	Type	Rôle
j	Entier	Compteur
i	Entier	Compteur
p	Entier	Variable provisoire

Algorithme

0) DEF PROC tri_insertion (var T : tab, n : entier)

1) Pour i de 2 à n faire

[p ← T [i], j ← i]

Tant que (j – 1 ≥ 1) et (p < T[j – 1]) faire

T [j] ← T [j – 1]
j ← j – 1

Fin Tant que

T [j] ← p

Fin pour

2) Fin tri_ insertion

b. Pascal (voir fichier : tri_inser.pas)

Leçon 2

Les algorithmes de recherche