

Les méthodes de tri

Sommaire

1) Tri par sélection	2
Principe.....	2
Analyse du module de tri par sélection (Tri en ordre croissant).....	2
Algorithme module de tri par sélection (Tri en ordre croissant)	2
Algorithme de la fonction minimum de la procédure triSelection (calcul de la position du plus petit élément dans un tableau à partir d'une position donnée)	2
Algorithme de la procédure permut de la procédure triSelection (permutation de deux valeurs en paramètre)	2
Programme Pascal du module de tri par sélection et son appel dans un programme permettant l'affichage du tableau avant et après le tri(Tri en ordre croissant)	3
2) Exercice 01 : application de tri par sélection décroissant	4

Les tableaux permettent de stocker plusieurs éléments de même type au sein d'une seule entité, Lorsque le type de ces éléments possède un ordre total, on peut donc les ranger en ordre croissant ou décroissant ainsi Trier un tableau c'est donc ranger les éléments d'un tableau en ordre croissant ou décroissant.

Il existe plusieurs méthodes de tri qui se différencient par leur complexité d'exécution et leur complexité de compréhension pour le programmeur.

1) Tri par sélection

Principe

- Trouver le plus petit élément et le mettre au début du tableau à trier
- Trouver le 2^{ème} élément le plus petit et le mettre en seconde position
- Trouver le 3^{ème} plus petit élément et le mettre à la 3^{ème} place
- Répéter la recherche du plus petit i^{ème} élément et le mettre à la i^{ème} position jusqu'à arriver au plus petit dernier élément

Analyse du module de tri par sélection (Tri en ordre croissant)

```
DEF PROC triSelection(var T : Tab ; n : entier)
  Résultat = T_trie
  pour i de 1 à n-1 faire
  [ mini ← minimum(t,n,i) ] si ( i <> mini) alors
                                permut(t[i],t[mini])
                                finSi
  finPour
Fin PROC triSelection
```

T.D.O.L

Objet	Type/Nature
T	Var/Tab
n	Var/entier
i	Var/entier
minimum	Fonction/entier
permut	Procédure

Algorithme module de tri par sélection (Tri en ordre croissant)

```
0) DEF PROC triSelection(var T : Tab ; n : entier)
1) pour i de 1 à n-1 faire
    [ mini ← minimum(t,n,i) ] si ( i <> mini) alors permut(t[i],t[mini]) finSi
    finPour
2) Fin PROC triSelection
```

Algorithme de la fonction minimum de la procédure triSelection (calcul de la position du plus petit élément dans un tableau à partir d'une position donnée)

```
0) DEF FN minimum(t:tab; n,m: entier):entier;
1) [ ppmin ← m ] pour j de m+1 à n faire
    si (t[j]<t[ppmin])alors ppmin ← j finSi
2) minimum ← ppmin
3) FIN minimum
```

Algorithme de la procédure permut de la procédure triSelection (permutation de deux valeurs en paramètre)

```
0) DEF PROC permut(var x , y : entier)
1) z ← x
2) x ← y
3) y ← z
4) FIN PROC permut
```

Programme Pascal du module de tri par sélection et son appel dans un programme permettant l'affichage du tableau avant et après le tri (Tri en ordre croissant)

```

Program applicationTriSelection;
Uses wincrt ;const max = 100;
Type tab = array[1..max]of integer ;
procedure remplirTab(var t : tab ; n : integer);
var i : integer ;
begin
  for i:=1 to n do
    begin
      write('t[,i,']=');readln(t[i]);
    end;
  end;
procedure affichageTab(t : tab ; n : integer);
var i : integer ;
begin
  for i:=1 to n do  write(t[i], ' | ');
end;
procedure triSelection (var t : tab ; n : integer);
var i,mini,j: integer;
procedure permut(var x , y : integer);
var z:integer;
begin
  z:=x; x:=y; y:=z;
end;
function minimum(t:tab;n,m:integer):integer;
var j,ppmin : integer;
begin
  ppmin :=m;
  for j := m+1 to n do
    if(t[j]<t[ppmin])then ppmin :=j;
  minimum := ppmin;
end;
begin
  for i := 1 to n-1 do
    begin
      mini:=minimum(t,n,i);
      if( i <> mini) then permut(t[i],t[mini]);
    end ;
  end;
var ta : tab ;
begin
  remplirTab(ta,4);
  writeln('tableau avant le tri : ');affichageTab(ta,4);triSelection(t,4); writeln(' ');
  writeln('tableau après le tri : ');
  affichageTab(ta,4);
end.

```

2) Exercice 01 : application de tri par sélection décroissant

```

program triDecroissant ;
uses wincrt ;
type tab = array[1..100]of integer ;
procedure saisieN(var n : integer);
begin
  repeat
    begin
      write('N = '); readln(n);
    end;
  until n in [5..15];
end;
procedure remplissage(var t : tab ; n : integer);
var i:integer;
begin
  randomize;
  for i:= 1 to n do
    begin
      t[i]:=10+ random(90);
    end;
end;
procedure affichage(t : tab ; n : integer);
var i : integer;
begin
  for i:= 1 to n do write(' | ',t[i]);
  writeln("");
end;
procedure triSelection (var t : tab ; n : integer);
var i,maxi,j: integer;
procedure permut(var x , y : integer);
var z:integer;
begin
  z:=x; x:=y; y:=z;
end;
function maximum(t:tab;n,m:integer):integer;
var j,ppmax : integer;
begin
  ppmax :=m;
  for j := m+1 to n do
    if(t[j]>t[ppmax])then ppmax :=j;
  maximum := ppmax;
end;
begin
  for i := 1 to n-1 do
    begin
      maxi:=maximum(t,n,i);
      if(t[i]<>t[maxi]) then permut(t[i],t[maxi]);
    end ;
  end;
var t : tab;n,i :integer;
begin
  saisieN(n); remplissage(t,n);
  writeln(' ----- T avant le tri ----- '); affichage(t,n);
  triSelection(t,n);
  writeln(' ----- T apres le tri ----- '); affichage(t,n);
end.

```