Tri baquets

Christian Jean CYRILLE

12 janvier 2020

1 Sujet Centrale 04

2 Corrigé

Opérations élémentaires : lire ou écrire dans une case mémoire - opérations sur les entiers (+,-,*,div,mod) - comparaisons d'entiers(<,>,>=,<=,< >)
Partie 1 - Tri d'entiers compris entre 0 et MAXVAL
const MAXDIM = 100 ; MAXVAL=1000 ;
type TABLE = array[0..MAXDIM] of INTEGER ;BACS = array[0..9] of TABLE ;

1 *instruction 1 : lecture au clavier des n éléments du tableau T*

```
for I := 1 to N do readln(T[I]) ;
```

$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5]$	$T[6]$	$T[7]$	$T[8]$	$T[9]$
6	4	2	8	4	2	3	6	4

instruction 2 : mise à zéro d'un tableau C d'occurrences pour chaque entier entre 0 et MAXVAL

```
for I := 1 to MAXVAL do C[I]:=0 ;
```

$C[0]$	$C[1]$	$C[2]$	$C[3]$	$C[4]$	$C[5]$	$C[6]$	$C[7]$	$C[8]$	$C[9]$	$C[10]$
0	0	0	0	0	0	0	0	0	0	0

instruction 3 : compter le nombre d'occurrences de chaque valeur T[I]

```
for I:= 1 to N do c[T[I]] := C[T[I]] + 1 ;
```

$C[0]$	$C[1]$	$C[2]$	$C[3]$	$C[4]$	$C[5]$	$C[6]$	$C[7]$	$C[8]$	$C[9]$	$C[10]$
0	0	2	1	3	0	2	0	1	0	0

instruction 4 :

```
COMPTEUR := 0;
for I:= 0 to MAXVAL do
begin
if C[I] < > 0 do begin
for J:=1 to C[I] do
begin
COMPTEUR := COMPTEUR + 1;
T[COMPTEUR] : = I;
end;
end;
end;
```

instruction 5 : affichage des éléments du Tableau T

```
for I := 1 to N do write(T[I]) ;
```

$T[1]$	$T[2]$	$T[3]$	$T[4]$	$T[5]$	$T[6]$	$T[7]$	$T[8]$	$T[9]$
2	2	3	4	4	4	6	6	8

Etude de la complexité

- instruction 1 : n lectures au clavier + n écritures dans les cases-mémoires de T
- instruction 2 : MAXVAL écritures dans les cases-mémoires de C
- instruction 3 : n cycles de 4 instructions(lecture en mémoire de T[I] ; lecture en mémoire de C[T[I]] ; addition C[T[I]] + 1 ; écriture pour l'affectation C[T[J]] :=C[T[J]]+1)
- instruction 4 : 1 écriture en mémoire COMPTEUR := 0
- instruction 5 : n lectures de T[I] pour affichage

Au pire, la complexité est :
$n + n + MAXVAL + 4n + 1 + (MAXVAL + 1) * (2n + 1)$
en $O(nMAXVAL + n)$

2
```
procedure tri_simple(T:TABLE);
var N, I, COMPTEUR : integer; C : TABLE;
begin
for I := 1 to MAXVAL do C[I]:=0 ;
for I:= 1 to N do C[T[I]] := C[T[I]] + 1 ;
COMPTEUR := 0;
for I:= 0 to MAXVAL do
begin
if C[I] < > 0 do begin
for J:=1 to C[I] do
begin
COMPTEUR := COMPTEUR + 1;
T[COMPTEUR] : = I;
end;
end;
end;
end;
```

Partie 2 - Gestion de tables

3
```
procedure vider(var T:TABLE);
begin
T[0]:= 0;
end;
```

4
```
procedure ajouter(var T:TABLE;p:integer);
begin
T[T[0] + 1]:= p;
T[0] := T[0] + 1;
end;
```

5
```
procedure concatener(vat T1: TABLE; T2 : TABLE);
var I :integer;
begin
for I:= 1 to T2[0] do begin
T1[T1[0] + I] :=T2[I]
end;
T1[0]:=T1[0] + T2[0]
end;
```

6 *Complexité de la fonction concaténer*
Dans la boucle for I :=1 to T2[0] il y a 3 instructions :

- 1 lecture T1[0]
- 1 calcul T1[0] + I
- 1 affectationT1[T1[0] + I] :=T2[I]

Cette boucle est suivie d'un calcul T1[0] :=T1[0] + T2[0] donc la complexité est $\boxed{3T2[0] + 1}$

7
```
function Max_Valeurs(T:table): integer;
var MAX : integer;
begin
if T[0] = 0 then Max_Valeurs:=-1
else
```

```
begin
MAX:=T[1];
for I := 2 to T[0] do if MAX < T[I] do MAX := T[I];
Max_Valeurs:=MAX
end;
end;
```

La complexité est au pire le Maximum entre 1 affectation dans le cas où T est vide et 1 + T[0]-1 + 1 c'est-à-dire $\boxed{max(1; T[0]+1)}$

8
```
function Nombre_Chiffres(p: word):word;
var C:word;
begin
C:=1;
while p div 10 > 0 do
begin
C:=C+1;
p:=p div 10
end;
Nombre_Chiffres := C;
end;
```

9
```
function Max_Chiffres(T:table): word;
var M : integer;
begin
if T[0] = 0 then Max_Chiffres := 0
else
begin
M := Max_Valeurs(T);
Max_Chiffres := Nombre_Chiffres(M)
end;
end;
```

10 Complexité de $MaxChiffres$ = complexité de $MaxValeurs$ + Complexité de$NombreChiffres$
$= n + maxc$ **Partie 3 - Tri d'entiers écrits en base 10 à l'aide de baquets**

11 r = 1
T 8 57 423 50 603 07 20 453 27
baquets[0] 2 50 20
baquets[3] 3 423 603 453
baquets[7] 3 57 7 27
T 8 50 20 423 603 453 57 07 27
r = 2
T 8 50 20 423 603 453 57 07 27
baquets[0] 2 603 07
baquets[2] 3 20 423 27
baquets[5] 3 50 453 57
T 8 603 07 20 423 27 50 453 57
r= 3
T 8 603 007 020 423 027 050 453 057
baquets[0] 5 007 020 027 050 057
baquets[4] 2 423 453
baquets[6] 1 603
T 8 7 20 27 50 57 423 453 603

12 La table est triée car on trie d'abord les n entiers : selon le chiffre des unités puis selon le chiffre des dizaines puis selon le chiffre des centaines, etc.....

13
```
procedure distribuer(var T : table, r : integer; baquets : BACS);
var I , P,, NC, Q : integer;
begin
for I := 1 to T[0] do
begin
p:=T[I];
```

```
NC:=Nombre_Chiffres(p);
if r > NC then ajoute(baquets[0],p)
else begin
q := puiss10(r);
j:=le chiffre en position r dans p;
ajoute(baquets[j],p)
end;
end;
end;
14 procedure Tri_Baquets(var T:Table);
var maxc, J, r : integer;
begin
maxc:=Max_Chiffres(T);
for r :=1 to maxc do
begin
for J:=0 to 9 do vider(baquets[J]);
distribuer(T,r,baquets)
end;
vider(T);
for J:=0 to 9 do concatener(T,baquets[J])
end;
```