

J. P. BENZÉCRI

## **Étude sous forme de problème d'un algorithme de tri écrit en langage ALGOL**

*Les cahiers de l'analyse des données*, tome 8, n° 1 (1983),  
p. 113-120

[http://www.numdam.org/item?id=CAD\\_1983\\_\\_8\\_1\\_113\\_0](http://www.numdam.org/item?id=CAD_1983__8_1_113_0)

© Les cahiers de l'analyse des données, Dunod, 1983, tous droits réservés.

L'accès aux archives de la revue « Les cahiers de l'analyse des données » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

ÉTUDE SOUS FORME DE PROBLÈME  
D'UN ALGORITHME DE TRI  
ÉCRIT EN LANGAGE ALGOL

[PROB. ALG. TRI]

par J. P. Benzécri (1)

1 Enoncé

L'objet de ce problème est d'étudier l'effet d'un algorithme écrit en langage ALGOL. Nous conseillons de lire attentivement le listage donné à la fin de l'énoncé avant de résoudre les questions proposées. Celles-ci sont, dans une large mesure indépendantes entre elles. Les questions 1, 2, 3 peuvent être abordées dans un ordre quelconque ; la question 4 présuppose seulement les résultats des questions 2 et 3.

1.1° On considère l'effet du programme suivant, constitué par les dernières instructions du listage ; en supposant que la fonction entière  $RN[N]$  prend une fois et une seule chacune des valeurs de l'intervalle  $[1 : CN]$  quand  $N$  parcourt cet intervalle.

```
NEQ := 0
pour N := 1 pas 1 jusqu'à CN faire
début
  W := si N = CN alors VN [N] + 1 sinon VN [N + 1] ;
  si VN [N] > W alors imprimer FAUX ;
  si VN [N] = W alors NEQ := NEQ + 1 ;
  si VN [N] < W alors
début
  NB := N - NEQ pas 1 jusqu'à N faire
  TRN [RN [NB]] := N - (NEQ/2) ;
  NEQ := 0
fin
fin
```

1.1°a A quelle condition le programme s'exécute-t-il sans jamais imprimer FAUX?

1.1°b On suppose qu'est réalisée la condition demandée en 1° - a), et que de plus, l'on n'a jamais  $W = VN[N]$ . Quelle relation y-a-t-il entre les fonctions  $RN$  et  $TRN$  définies sur l'intervalle  $[1 : CN]$  ?

---

(1) Professeur de statistique. Université Pierre et Marie Curie;

1.1°c On considère les données suivantes (pour CN = 7)

N	1	2	3	4	5	6	7
VN	0,6	0,7	0,7	0,7	0,8	1,2	1,2
RN	6	4	5	7	3	1	2

Quel est alors le contenu du tableau TRN après exécution du programme :

1.1°d Expliquer quel est, en général, l'effet du programme.

2° - Dans cette question on étudie l'effet d'une partie du programme, en la simplifiant par l'élimination de tout ce qui concerne les tableaux RN et TRN. Soit donc le programme :

```

N := NA := INA ; NB := INA + LA ;
DEB ;
si INA + LA < NA alors aller à ETB ;
si INA + LA + LB < NB alors aller à ETA ;
si VN [NA] < VN [NB] alors aller à ETA sinon aller à ETB ;
ETA ;
TVN [N] := VN [NA]
NA := NA + 1 ; N := N + 1 ; aller à FIN ;
ETB ;
TVN [N] := VN [NB] ;
NB := NB + 1 ; N := N + 1 ;
FIN ;
si N < INA + LA + LB alors aller à DEB
pour N := INA pas 1 jusqu'à INA + LA + LB - 1 faire
    VN [N] := TVN [N] ;

```

2° - a) Préciser en fonction de INA, LA et LB quelles sont les valeurs de NA ou NB pour lesquelles on doit calculer VN [NA] ou VN [NB]

2°- b) déduire de 2° - a) une relation qui doit exister entre INA, LA, LB et CN pour que le programme soit exécutable.

2° - c) On suppose que initialement les deux suites des nombres ci-dessous satisfont aux inégalités indiquées.

$$\begin{aligned} &VN [INA] \leq VN [INA + 1] \leq \dots \leq VN [INA + LA - 1] ; \\ &VN [INA + LA] \leq VN [INA + LA + 1] \leq \dots \leq VN [INA + LA + LB - 1] \end{aligned}$$

Que peut on dire de la suite des valeurs VN [N] pour N variant de INA à INA + LA + LB - 1, après exécution du programme ?

3° - Dans cette question on étudie sur un cas particulier l'effet du programme obtenu en supprimant du listage le bloc des dernières instructions (objet de la question 1) et de plus en simplifiant, (comme dans la question 2) par élimination de tout ce qui concerne les tableaux RN et TRN. On a donc le programme

```
lire réel tableau VN [1 : CN] ;
LA := 1 ; INA := 1
DZZ ;
LB := inf (LA, CN - (INA + LA - 1))
N := NA := INA ; NB := INA + LA
```

.....

(la suite des instructions est à prendre dans la question 2)

.....

```
INA := INA + LA + LB ;
si CN < INA + LA alors
début
  LA := 2 * LA ; INA := 1 fin
si INA + LA ≤ CN alors aller à DZZ ;
```

3° - a) On considère le tableau de données suivant (CN = 7) :

N	1	2	3	4	5	6	7
VN	1,2	1,2	0,8	0,7	0,7	0,6	0,7

Combien de fois passe-t-on par l'étiquette DZZ avec pour LA sa valeur initiale LA = 1 ? Quel est l'état du tableau VN lorsqu'on passe pour la première fois à l'étiquette DZZ avec une valeur de LA différente de 1.

3° - b) Quelles sont les valeurs différentes que prend le nombre entier LA au passage de l'étiquette DZZ ; préciser combien de fois on passe par l'étiquette DZZ avec chacune de ces valeurs.

3° - c) Quel est l'état du tableau VN après exécution du programme (objet de la question 3°) sur les données proposées en 3° - a).

4° - Expliquer l'effet du programme du listage (avec exécution de ce qui concerne RN et TRN) jusqu'à l'instruction NEQ := 0 exclue. On pourra prendre pour exemple le cas où CN = 7 avec le tableau initial VN proposé en 3° - a). Quel est, en fonction de CN, l'ordre de grandeur du temps de rotation de l'algorithme.

5° - Quel est l'effet de l'ensemble du programme donné sur le listage.

LISTAGE de l'ALGORITHME

```

réel W ;
entier CN, N, NA, NB, LA, LB, INA, NEQ ;
étiquette DZZ, DEB, FIN, ETA, ETB,
réel tableau VN, TVN, TRN[1:CN];
entier tableau RN[1:CN]
début
lire réel tableau VN [1 : CN] ;
pour N := 1 pas 1 jusqu'à CN faire RN[N] := N
LA := 1 ; INA := 1 ;
DZZ ;
LB := inf (LA, CN - (INA + LA - 1)) ;
N := NA := INA ; NB := INA + LA ;
DEB
si INA + LA ≤ NA alors aller ETB ;
si INA + LA + LB ≤ NB alors aller à ETA ;
si VN [NA] ≤ VN [NB] alors aller à ETA sinon aller à ETB ;
ETA ;
TVN [N] := VN [NA] ; TRN [N] := RN [NA] ;
NA := NA + 1 ; N := N + 1 ; aller à FIN
ETB ;
TVN [N] := VN [NB] ; TRN [N] := RN [NB] ;
NB := NB + 1 ; N := N + 1 ;
FIN
si N < INA + LA + LB alors aller à DEB
pour N := INA pas 1 jusqu'à INA + LA + LB - 1 faire
début VN [N] := TVN [N] ; RN [N] := TRN [N] fin ;
INA := INA + LA + LB ;
si CN < INA + LA alors
début LA := 2 * LA ; INA := 1 fin ;
si INA + LA ≤ CN alors aller à DZZ ;

NEQ := 0
pour N := 1 pas 1 jusqu'à CN faire
début W := si N = CN alors VN [N] + 1 sinon VN [N + 1] ;
si VN [N] > W alors imprimer FAUX ;
si VN [N] = W alors NEQ := NEQ + 1 ;

```

.../...

```

si VN [N] < W alors
début
pour NB : = N - NEQ pas 1 jusqu'à N faire
    TRN [RN [NB]] : = N - (NEQ/2) ;
    NEQ : = 0
fin
fin
fin

```

## 2 Solution

2.1°a : Si  $N = CN$ ,  $W$  est nécessairement supérieur à  $VN[N]$  et il n'y a pas lieu d'imprimer FAUX ; pour  $N < CN$ , on imprime FAUX si  $VN[N] > VN[N+1]$  ; en somme le programme s'exécute sans jamais imprimer FAUX à la condition que la suite  $\{VN[1], VN[2], \dots, VN[CN]\}$  soit non-décroissante.

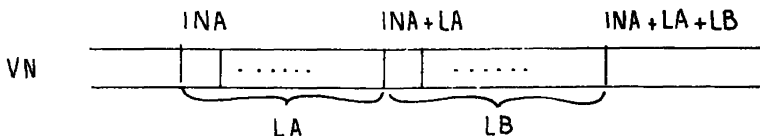
2.1°b : La condition supplémentaire de b,  $W \neq VN[N]$ , combinée à celle de a, impose que la suite  $\{VN[1], VN[2], \dots, VN[N]\}$  soit strictement croissante. Dans ce cas on a, tout au long de l'exécution du programme,  $NEQ = 0$  ; donc  $NB = N$  ; et  $TRN[RN[N]] = N$ . On a posé au départ que la fonction RN réalise une correspondance biunivoque de l'intervalle entier  $[1:CN]$  avec lui-même ; ici la fonction TRN réalise la correspondance inverse de RN.

2.1°c : On a le résultat suivant, dont le principe sera expliqué en d :

N	1	2	3	4	5	6	7
VN	0,6	0,7	0,7	0,7	0,8	1,2	1,2
RN	6	5	4	7	3	1	2
TRN	6,5	6,5	5	3	3	1	3

2.1°d : On a vu en a, que le programme vérifie que la suite  $\{VN[1], \dots, VN[CN]\}$  est non décroissante (i.e. que tout terme est supérieur ou égal à son prédécesseur immédiat) ; ceci étant réalisé, on lit la suite des  $VN[N]$ , en notant sur le compteur NEQ le nombre (diminué d'une unité) des termes consécutifs égaux : si la suite est strictement croissante (cf. b) le compteur NEQ reste à 0 ; dans le cas de l'exemple (cf. c) il atteint une fois la valeur 2, et une fois la valeur 1. On attribue alors aux termes consécutifs égaux un rang moyen, qui n'est autre que celui du dernier diminué de  $NEQ/2$ . Ceci fait ce rang est inscrit au tableau TRN. Par exemple le terme ayant pour indices  $RN = 7$  et  $N = 4$ , a pour valeur 0,7 et pour rang 3 (car 0,7 est la valeur commune au 2-ème, 3-ème et 4-ème nombres de la suite des  $VN[N]$ ) ; on attribue donc à l'indice 7 le rang  $TRN[7] = 3$ .

2.2°a : Les indices  $N$ ,  $NA$ ,  $NB$  ont pour valeurs initiales respectives :



INA, INA et INA+LA ; à chaque passage par l'étiquette DEB, l'indice N augmente de 1, ainsi que l'un des deux indices NA ou NB ; le choix étant fait d'après trois instructions en *si* successives. Le retour à l'étiquette DEB ne se fait que si  $N <_{s} \text{INA} + \text{LA} + \text{LB}$ . Donc N varie de INA à INA+LA+LB-1 ; soit une variation de (LA+LB). Quant aux indices NA et NB ils parcourent respectivement les intervalles [INA:INA+LA-1] et [INA+LA:INA+LA+LB-1] : intervalles d'amplitudes LA et LB.

2.2°b : Le programme s'exécutera à condition que les indices N, NA, NB ne sortent pas du domaine déclaré pour les tableaux ; d'où l'inégalité :

$$\text{INA} + \text{LA} + \text{LB} - 1 \leq \text{CN}.$$

2.2°c : Le programme recopie d'abord les deux suites croissantes

{VN[INA], ..., VN[INA+LA-1]}, et :

{VN[INA+LA], ..., VN[INA+LA+LB-1]},

en les interclassant dans le tableau TVN entre les positions INA et INA+LA+LB-1 ; on recopie ensuite la suite croissante TVN ainsi obtenue, dans le tableau VN. La suite des valeurs {VN[INA], ..., VN[INA+LA+LB-1]} est alors devenue une suite croissante.

2.3°a : La partie du programme étudiée dans la question 2°, opère sur deux segments consécutifs de longueurs respectives LA et LB et débutant à la valeur INA de l'indice N. Dans cette troisième question on considère l'enchaînement des systèmes de valeurs (INA, LA, LB) définissant les domaines à interclasser ; la définition d'un nouveau système correspondant au passage par l'étiquette DZZ.

Le programme opérant sur les données proposées passe trois fois par l'étiquette DZZ avec la valeur LA = 1 : ce qui correspond aux triplets

$$(\text{INA}, \text{LA}, \text{LB}) = (1, 1, 1) ; (3, 1, 1) ; (5, 1, 1).$$

Au premier passage on interclasse les valeurs VN[1] et VN[2] : c'est-à-dire qu'on permute ces valeurs à supposer qu'elles ne soient pas rangées dans l'ordre  $\text{VN}[1] \leq \text{VN}[2]$  : en fait ici  $\text{VN}[1] = \text{VN}[2]$  ; il n'y a rien à changer. Au deuxième passage on fait de même pour VN[3] et VN[4] qu'il faut permuter. Au troisième passage on permute de même VN[5] et VN[6]. Il reste une valeur isolée VN[7] sur laquelle il n'y a rien à faire : on pose donc LA = 2 pour entreprendre d'interclasser deux à deux les segments successifs de longueur 2 du tableau VN : l'état de ce tableau est alors :

N	1	2	3	4	5	6	7
VN	1,2	1,2	0,7	0,8	0,6	0,7	0,7

2.3°b : On interclasse les segments [1:2] et [3:4] ; puis les segments [5:6] et [7] (ce dernier réduit à un seul élément) ; ceci correspond aux deux triplets  $(\text{INA}, \text{LA}, \text{LB}) = (1, 2, 2) ; (5, 2, 1)$ .

Il reste alors à interclasser les segments [1:4] et [5:7] ; ce qui correspond à un seul passage par l'étiquette DZZ avec le triplet  $(\text{INA}, \text{LA}, \text{LB}) = (1, 4, 3)$ .

2.3°c : Le classement des valeurs du tableau VN est alors achevé , on a :

N	1	2	3	4	5	6	7
VN	0,6	0,7	0,7	0,7	0,8	1,2	1,2

2.4° : On considère maintenant l'effet du programme, non comprise la partie terminale objet de la question 1° ; on reprend donc les résultats des questions 2° et 3°, mais en tenant compte des tableaux TRN et RN. En bref on peut considérer que le tableau RN donne un indice; et que le tableau VN donne des valeurs correspondant à cet indice : au départ on a  $RN[N] = N$  ; et  $VN[N]$  n'est autre que la valeur correspondant à l'indice N. Mais au cours des permutations que le programme opère sur les valeurs VN il faut que soient simultanément permuées les valeurs de l'indice RN. Ainsi après trois passages par l'étiquette DZZ on a, comme l'indique la question 2°a :

N	1	2	3	4	5	6	7
VN	1,2	1,2	0,7	0,8	0,6	0,7	0,7
RN	1	2	4	3	6	5	7

De même qu'on a permuté entre elles les paires de valeurs (VN[3], VN[4]) et (VN[5],VN[6]) sont permuées dans RN les paires d'indices (3,4) et (5,6).

Suivent deux passages par l'étiquette DZZ avec LA = 2 (cf. 3°b) :

N	1	2	3	4	5	6	7
VN	0,7	0,8	1,2	1,2	0,6	0,7	0,7
RN	4	3	1	2	6	5	7

Enfin après un dernier passage avec LA = 4 (cf. 3°c) il vient :

N	1	2	3	4	5	6	7
VN	0,6	0,7	0,7	0,7	0,8	1,2	1,2
RN	6	4	5	7	3	1	2

Ainsi on a rangé dans l'ordre croissant les valeurs contenues dans le tableau VN ; en conservant dans le tableau RN le rang qu'avait initialement cette valeur. On notera que la valeur 0,7 qui se rencontre trois fois figure avec les trois valeurs 4, 5, 7 de RN ; ce qui correspond aux rangs des trois occurrences de 0,7 dans la suite initiale.

Quant au temps d'exécution on peut dire en bref qu'à chaque valeur de LA (LA = 1 ; puis LA = 2 etc.) il correspond une lecture complète des tableaux avec des opérations dont la longueur est proportionnelle à CN. Le nombre des lectures est fixé par la suite des valeurs de LA : ici 1, 2, 4 et en général une progression géométrique de raison 2 dont le dernier terme est compris dans l'intervalle  $[CN/2 : CN-1]$ . Donc un ordre de grandeur de  $\log_2(CN)$  pour le nombre des valeurs de LA utilisées ; et pour la durée d'exécution du programme :  $CN \cdot \log_2(CN)$ .



5°) Comme on l'a expliqué en 1° le programme s'achève en donnant pour chaque valeur son rang exprimé en fonction de son rang initial ; lequel est conservé tout au long du programme comme valeur de RN . Eventuellement (cf. 1°d) quand plusieurs valeurs sont au même rang on affecte à chacune des occurrences un rang unique calculé comme une valeur moyenne.

L'algorithme de tri présenté ici est à notre connaissance le plus rapide ; il est d'une grande utilité en analyse des données , notamment pour découper une variable en classes , effectuer un codage par rangs etc. .