

5 SORT

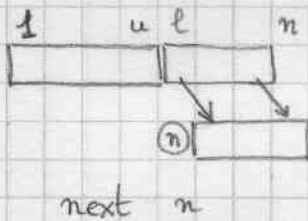
entrée valeurs  $K_1 \dots K_{N_0} K_{N_0+1} \dots K_N$   
 déjà triés  
 $P(1) \dots P(N_0)$   
 permutable de  $1 \dots N_0$  ( $N_0 \in [0, N]$ )  
 $K_{P(1)} \leq K_{P(2)} \leq \dots \leq K_{P(N_0)}$

sortie  $P(1) \dots P(N)$   
 per de  $1 \dots N$   
 $K_{P(1)} \leq K_{P(2)} \dots \leq K_{P(N)}$   
 $K_{P(i)} = K_{P(j)} \Rightarrow i \leq j$

pour  $n = N_0 + 1, N$  [ret si  $N_0 = N$ ]  
 si  $n \neq 1$   $P(1) \dots P(n-1)$  est une permutable de  $1 \dots n-1$

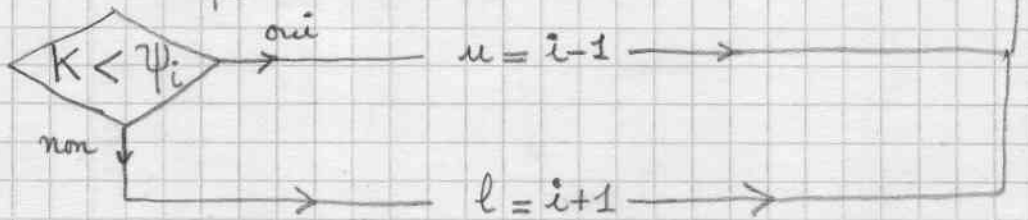
$K = K_n$   
 $l = 1$   
 $u = n - 1$   
 ici  $\Psi_{l-1} \leq K < \Psi_{u+1}$   
 $\Psi_\alpha$  désigne  $K_{P(\alpha)}$   
 avec  $\Psi_n = \infty$   
 $\Psi_0 = -\infty$

si  $u < l$  (on a alors  $u = l - 1$  et  $\Psi_u \leq K < \Psi_l$ )



sinon  $i = \lfloor \frac{l+u}{2} \rfloor$

Comparer  $\Psi_i$  et  $K$



N	index * 32	variable	chaines
200	0	1	0
1000	2	8	5
2000	7	21	13
5000	37		
10000	139		

5) soit  $S(k_1, k_2, \dots), N, P, P(k'_1, \dots, k'_N) [N_0]$

entrée  $S, N, P$ : définit  $N$  éléments (index  $\times 32$ , variable ou chaîne)  $S_1 \dots S_N$   
 $N > 0$

$P$ : index  $\times 32$

$N_0 \in [0, N-1]$  les éléments  $S_1 \dots S_{N_0}$  sont déjà triés

sortie met dans  $P(1) \dots P(N)$  la perm. de  $1 \dots N$  rangant  $S_1 \dots S_N$

YSORT: BSR WTSRC (3)142 décode  $S, N, P$

BSR CMPDET (5)262a

MOVEM.L <sup>incr N</sup> D0/D1/A3, -(SP)

~~MOVE.L A0, GEG55~~ (X) (?)

BSR DECCRV (5)

MOVE.L 4(SP), D1 N

BSR WTPER (3)144  $P(k'_1, \dots, k'_N)$  index  $\times 32 \rightarrow$  A3 adresse D1 incluse = N

SUBQ #4, A3

MOVE.L A3, -(SP)

MOVEQ #0, D3

BSR DECCRV

BNE GEK10

BSR WADR

BSR WVGAD (X)

~~GEK10: MOVE.L 8(SP), D1 N~~

~~CMP.L D1, D3~~

~~BHI ERR1XH~~

~~MOVE.L D3, -(SP)~~  $\rightarrow$  hors du tableau si  $N_0 \notin [0, N]$

~~MOVE.L #GEK15, DICHOTR~~ ] retour de DICHOT

~~BEQ GEK12~~

~~MOVEM.L (SP), D1/A3~~

~~ADDQ #4, A3~~

~~BSR XPEROK (3)145~~

} Cas  $N_0 \neq 0$ : vérifier que  $P(1) \dots P(N_0)$  est une perm. de  $1 \dots N_0$

index	variable/chaîne
0	incrément sur adr
4	N
8	adresse de $S_1$

X

YSORT: SUB #92, SP

BSR WTSRC (3) 142 décode S, N, P

BSR CMPDET (5) 262a initialise et décode P(k...)

SUBQ #4, A3

MOVE.L A3, -(SP)

~~MOVE.L #0, D3~~

~~BSR WVGARZ~~

} d3 ~~avec~~  
= [No] (défaut = 0)

MOVE.L 8(SP), D1 N

CMP.L D1, D3

BHI ERRIXM → hors du tableau

MOVE.L #GEEK15, 28(SP) adresse de retour de DICHOT

MOVE.L D3, -(SP)

BEQ GAS50

MOVEM.L (SP), D1/A3

cas No ≠ 0 P(1) ... P(No)  
doit être une permutation de 1...No

ADDQ #4, A3

BSR XPEROK

GAS50: BSR WTSCM (9) 265.2 décode [ , fcmf ]

augmenter n

```

GEK12: MOVE.L (SP), D5
      ADDQ.L #1, (SP)
      CMP.L R(SP), D5
      BLT GEK14
      ADD #26#100, SP
      RTS
  
```

$D5 = n - 1 \quad [ = u ]$

n
4 adr P(0)
8 S
12 N
16 adr/n°

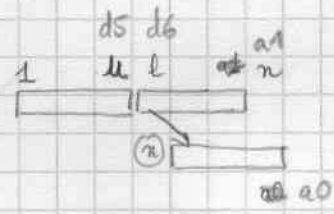
→ ok  
↓ n > N

⊗

```

GEK14: MOVEQ #1, D6
      BRA DICHOT
  
```

$D6 = l$



$D5 = D6$   
 $u = l - 1$  et  $\forall u \leq k < l$

retour de DICHOT

```

GEK15: MOVEM.L (SP), D0/A0
      SUB.L D0, D6
      ADD.L D0, D0
      ADD.L D0, D0
      ADD.L D0, A0
      LEA #4(A0), A0
      BRA GEK17
  
```

- nb d'éléments = D6  
à déplacer

```

GEK16: MOVE.L -(A1), -(A0)
GEK17: ADDQ.L #1, D6
      BLE GEK16
      MOVE.L (SP), (A1)
      BRA GEK12
  
```

⊗





n
adr P(0)
δ
N
adr/n

```

DICHOT: CMP.L D6, D5
        BCC GEK20
        JMP GEK15.L
DICHOT = * - 4
    ) return
    MOVE.L 32(SP), A0
    JMP (A0)
  
```

```

GEK20: MOVE.L D5, D4
        ADD.L D6, D4
        ASR.L #1, D4
  
```

$$D4 = i = \lfloor \frac{l+u}{2} \rfloor$$

```

MOVEM.L (SP), D1/A0
        MOVE.L D4, D0
        ADD.L D0, D0
        ADD.L D0, D0
        MOVE.L (A0, D0.L), D0
        SUBQ.L #1, D0
        SUBQ.L #1, D1
  
```

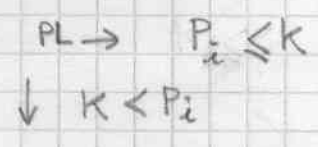
$$D0 = P_i$$

```

        JSR 0.L
DICHOT = * - 4
  
```

```

    MOVE.L 28(SP), A0
    JSR (A0)
  } compare KD0 et KD1
    Pi
  
```



```

BPL GEK22
MOVE.L D4, D5
SUBQ.L #1, D5
BRA DICHOT
  
```

```

GEK22: MOVE.L D4, D6
        ADDQ.L #1, D6
        BRA DICHOT
  
```