TD – Tris récursifs

1. Pile - Tri fusion (mergesort)

t : 6 3 0 9 1 7

Simulez la pile sur l'appel `trier-fusion(t,1,6)` en donnant aussi les valeurs intermédiaires de `t` après l'appel de fusionner.

```
Procédure trier-fusion (E/S t:tab, E inf,sup:entier)
Var m : entier
Début
  si inf=sup
    alors écrire(''le tableau est trié'')
    sinon
      m←(inf+sup) div 2
      trier-fusion(t,inf,m)   {@1}
      trier-fusion(t,m+1,sup) {@2}
      fusionner(t,inf,m,sup)
  finSi
Fin

Procédure fusionner (E/S t:tab, E d,m,f:entier)
Var i,j,k : entier
    temp : tab
Début
  i←d
  j←m+1
  Pour k←1 à f-d+1 inc +1 Faire
    Si i≤m et j≤f
       Alors Si t[i]≤t[j]
                  Alors
                     temp[k]←t[i]
                     i←i+1
                  Sinon
                     temp[k]←t[j]
                     j←j+1
             FinSi
       Sinon Si i≤m Alors
                      temp[k]←t[i]
                      i←i+1
                    Sinon
                      temp[k] ←t[j]
                      j←j+1
             FinSi
    FinSi
  FinPour
  Pour k←1 à f-d+1 inc +1 Faire
     t[d+k-1]←temp[k]
  FinPour
Fin
```

~~@2 t = 0 3 6 1 9 7 i=6 s=6~~
~~@2 t = 0 3 6 9 1 7 i=5 s=5~~
~~@1 t = 0 3 6 9 1 7 i=4 s=4~~
~~@1 t = 0 3 6 9 1 7 i=4 s=5 m=4~~
~~@2 t = 0 3 6 9 1 7 i=4 s=6 m=5~~
~~@2 t = 3 6 0 9 1 7 i=3 s=3~~
~~@2 t = 6 3 0 9 1 7 i=2 s=2~~
~~@1 t = 6 3 0 9 1 7 i=1 s=1~~
~~@1 t = 6 3 0 9 1 7 i=1 s=2 m=1~~
~~@1 t = 6 3 0 9 1 7 i=1 s=3 m=2~~
~~@0 t = 6 3 0 9 1 7 i=1 s=6 m=3~~

1. Fusionner(t,d=1,m=1,f=2) et t = 6 3 0 9 1 7
➔ 3 6 0 9 1 7

2. Fusionner(t,d=1,m=2,f=3) et t = 3 6 0 9 1 7
➔ 0 3 6 9 1 7

3. Fusionner(t,d=4,m=4,f=5) et t = 0 3 6 9 1 7
➔ 0 3 6 1 9 7

4. Fusionner(t,d=4,m=5,f=6) et t = 0 3 6 1 9 7
➔ 0 3 6 1 7 9

5. Fusionner(t,d=1,m=3,f=6) et t = 0 3 6 1 7 9
➔ 0 1 3 6 7 9

2. Pile - Tri rapide (quicksort)

t :	4	10	8	18	12	2	16	0	14	6

Simuler la pile sur l'appel `tri-rapide(t,1,10)` {@0} en donnant aussi les valeurs intermédiaires de t après l'appel de `partionner`.

```
Const max = 100
Type tab = tableau [1...max] d'entier

Procédure tri-rapide(E/S t : tab, E inf,sup : entier)
Var p : entier
Début
  si inf<sup
     Alors partionner(t,inf,sup,p)
           tri-rapide(t,inf,p-1){@1}
           tri-rapide(t,p+1,sup){@2}
  Finsi
Fin

Procédure partitionner (E/S t : tab, E d,f : entier, S p : entier)
Var i,j,pivot : entier
Début
   pivot←t[d]
   i←d
   j←f
   TantQue i≤j Faire
      TantQue t[i]≤pivot et i≤j Faire
         i←i+1
      FinTantQue
      TantQue t[j]>pivot et i≤j Faire
         j←j-1
      FinTantQue
      Si i≤j alors échanger(t[i],t[j])
      FinSi
   FinTantQue
   p←j
   échanger(t[d],t[j])
Fin
```

t 0:	**4**	0	2	18	12	8	16	10	14	6
t 1:	2	0	**4**	18	12	8	16	10	14	6
t 2:	**0**	2	**4**	18	12	8	16	10	14	6
t 3:	**0**	2	**4**	**6**	12	8	10	16	14	**18**
t 4:	**0**	2	**4**	**6**	**10**	8	**12**	16	14	**18**
t 5:	**0**	2	**4**	**6**	**8**	10	**12**	**16**	14	**18**
t 6:	**0**	2	**4**	**6**	**8**	10	**12**	**14**	16	**18**

~~@2 t6 inf=11 sup=10~~
~~@2 t6 inf=10 sup=9~~
~~@1 t6 inf=8 sup=8~~
~~@2 t5 inf=8 sup=9 p=9~~
~~@2 t5 inf=7 sup=6~~
~~@1 t5 inf=5 sup=5~~
~~@1 t4 inf=5 sup=6 p=6~~
~~@2 t4 inf=5 sup=9 p=7~~
~~@1 t3 inf=4 sup=3~~
~~@1 t3 inf=4 sup=9 p=4~~
~~@2 t2 inf=4 sup=10 p=10~~
~~@2 t2 inf=3 sup=2~~
~~@1 t2 inf=1 sup=1~~
~~@1 t1 inf=1 sup=2 p=2~~
~~@0 t0 inf=1 sup=10 p=3~~