

## 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. 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 partitionner.

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 partitionner(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