TD1 - CORRECTION Exercice 1

Tri par insertion : étapes d’évolution de la liste de nombres

tri\_insertion(L) avec L=[10,3,7,5,9,7,8,0,8]

i=1 et j=1 les éléments comparés L[1]=3 et L[0]=10, on les échange

[**3**, **10**, 7, 5, 9, 7, 8, 0, 8]

i=2 et j=2 les éléments comparés L[1]=10 et L[2]=7, on les échange

[3, **7**, **10**, 5, 9, 7, 8, 0, 8]

i=3, L[3]=5, on échange L[3] et L[2] puis L[2] avec L[1]

[3, 5, 7, 10, 9, 7, 8, 0, 8]

i=4 ; L[4]=9 comparé au précédent. On les échange.

[3, 5, 7, **9**, **10**, 7, 8, 0, 8]

i=5 ; L[5]=7 comparé aux précédents

[3, 5, 7, **7**, **9**, **10**, 8, 0, 8]

i=6 ; L[6]=8 comparé aux précédents

[3, 5, 7, 7, **8**, **9**, **10**, 0, 8]

i=7 ; L[7]=0 comparé aux précédents

[**0**, **3**, **5**, **7**, **7**, **8**, **9**, **10**, 8]

i=8 ; L[8]=8 comparé au précédent

[0, 3, 5, 7, 7, 8, **8**, **9**, **10**]

Tri rapide « quicksort » : étapes d’évolution de la liste de nombres

tri\_quicksort(L,0,8) avec L=[10,3,7,5,9,7,8,0,8]

segmente(L,0,8) ; g=1 ; d=8 ; p=L[0]=10

d=8 et g=8 ; k=8 et on échange L[0] et L[8]

d’où k=8 et [**8**, 3, 7, 5, 9, 7, 8, 0, **10**]

On appelle tri\_quicksort(L,0,7) et tri\_quicksort(L,9,8) :

tri\_quicksort(L,9,8) ; 9<8 non vérifié : on sort de la boucle

tri\_quicksort(L,0,7) appelle :

segmente(L,0,7) ; g=1 ; d=7 et p=L[0]=8

d=7 car L[0]<p ; g=4 : on échange L[4] et L[7]

[8, 3, 7, 5, **0**, 7, 8, **9**, 10] ; d=6 et g=5

g=6 ; k=6 et on échange L[0] et L[6]

[**8**, 3, 7, 5, 0, 7, **8**, 9, 10] (visuellement ne change rien)

tri\_quicksort(L,0,5) et tri\_quicksort(L,7,7)

tri\_quicksort(L,7,7) ; g=8 et d=7 on sort de la boucle

tri\_quicksort(L,0,5) appelle :

segmente(L,0,5) ; g=1 ; d=5 et p=L[0]=8

d=5 ; g=5 ; d=g donc k=5

on échange L[0] et L[5] : [**7**, 3, 7, 5, 0, **8**, 8, 9, 10]

tri\_quicksort(L,0,4) et tri\_quicksort(L,5,5)

tri\_quicksort(L,5,5) ; g=6 et d=5 on sort de la boucle

tri\_quicksort(L,0,4) appelle :

segmente(L,0,4) ; g=1 ; d=4 et p=L[0]=7

d=4 ; g=5 ; d<g on échange L[0] et L[4]

k=4 ; [**0**, 3, 7, 5, **7**, 8, 8, 9, 10]

tri\_quicksort(L,0,3) et tri\_quicksort(L,4,4)

tri\_quicksort(L,4,4), on sort de la boucle

tri\_quicksort(L,0,3) appelle :

segmente(L,0,3) ; g=1 ; d=3 et p=L[0]=0

k=0 et [0, 3, 7, 5, 7, 8, 8, 9, 10]

tri\_quicksort(L,0,-1) et g=1 et d=-1, on sort de la boucle

tri\_quicksort(L,1,3) appelle :

segmente(L,1,3) ; g=2 et d=3 avec p=L[1]=3,

d=1 on échange L[1] et L[1] ;

k=1 et [0, **3**, 7, 5, 7, 8, 8, 9, 10]

tri\_quicksort(L,1,0) (on sort de la boucle) et tri\_quicksort(L,2,3)

segmente(L,2,3)

k=3 et [0, 3, **5**, **7**, 7, 8, 8, 9, 10]

tri\_quicksort(L,2,2) et tri\_quicksort(L,4,3)

On sort définitivement de la boucle de récursivité

Tri fusion : étapes d’évolution de la liste de nombres

tri\_fusion(L,0,8)

L=[10,3,7,5,9,7,8,0,8]

tri\_fusion(L,0,8) ; m=4 ; tri\_fusion(L,0,4) et tri\_fusion(L,5,8) ; fusion\_listes(tab,0,8,4)

n1=5 ; n2=4 ; G=[10,3,7,5,9] et D=[7,8,0,8] ; k=0 ; i=0 ; 10>7 ; [**7**,3,7,5,9,7,8,0,8] ; j=1 ; k=1 ; i=0 ;10>8 ; [**7**,**8**,7,5,9,7,8,0,8] ; j=2 ; k=2 ; i=0 ; [**7**,**8**,**0**,5,9,7,8,0,8] ; j=3 ; k=3 ; [**7**,**8**,**0**,**8**,9,7,8,0,8] ; j=4 ; k=4 ;

[3, 10, 7, 5, 9, 7, 8, 0, 8]

[3]

[]

[3, 7, 10, 5, 9, 7, 8, 0, 8]

[3, 7]

[]

[]

[3, 7, 10, 5, 9, 7, 8, 0, 8]

[5]

[3, 5, 7, 9, 10, 7, 8, 0, 8]

[3, 5, 7, 9]

[]

[]

[3, 5, 7, 9, 10, 7, 8, 0, 8]

[7]

[]

[]

[3, 5, 7, 9, 10, 7, 8, 0, 8]

[0]

[3, 5, 7, 9, 10, 0, 7, 8, 8]

[0, 7, 8]

[0, 3, 5, 7, 7, 8, 8, 9, 10]