Université de Relizane Année Universitaire 2020/2021

Département d’Informatique 2 ime année informatique

Module : ASD3

**FICHE TD N°2**

**Exercice 01**

Donner l’algorithme de tri à bulle.

**Exercice 02**

Donner la procédure récursif « **Tri-Insertion »** qui permet de trier un tableau d’entier d’une manière récursive.

**Exercice 03**

**Tri rapide :**

La méthode consiste à placer un élément du tableau (appelé pivot) à sa place définitive, en permutant tous les éléments de telle sorte que tous ceux qui sont inférieurs au pivot soient à sa gauche et que tous ceux qui sont supérieurs au pivot soient à sa droite.

Cette opération s'appelle le partitionnement. Pour chacun des sous-tableaux, on définit un nouveau pivot et on répète l'opération de partitionnement. Ce processus est répété récursivement, jusqu'à ce que l'ensemble des éléments soit trié.

Concrètement, pour partitionner un sous-tableau :

* le pivot est placé à la fin (arbitrairement), en l'échangeant avec le dernier élément du sous-tableau ;
* tous les éléments inférieurs au pivot sont placés en début du sous-tableau ;
* le pivot est déplacé à la fin des éléments déplacés.

**Travail demandé :**

* Ecrire la procédure partitionner qui permet de partitionner le tableau en 2.
* Ecrire la procédure tri rapide.

**Solution :**

**Exercice 01**

 **L’algorithme de tri à bulle.**

Algorithme tri\_à\_bulle ;

Var

T[1..N] tableau d’entiers ;

Nb : entier ;

I,j,val :entier ;

Début

Pour i allant de nb-1 à 1 par pas de -1 faire

Pour j allant de 1 à i faire

Si (T[j-1]> T[j]) alors

Val T[j];

T[j] T[j-1] ;

T[j-1] val ;

Fin si ;

Fin pour ;

Fin pour ;

Fin.

**Exercice 02**

**procédure** **Tri-Insertion** ( A : tableau d’entier , n : entier) ;

 debut

 si (n>0)

**alors**

 **Tri-Insertion**(A,n-1) ;

 x ← A[n] ;

 j ← n-1 ;

 **tant que (** j >= 0) **et**  (A[j] > x) faire

 A[j+1] ← A[j] ;

 j ← j-1 ;

 **fin tant que ;**

 A[j+1] ← x ;

 **Fin si ;**

 **Fin procédure ;**

**Exercice 3 :**

Procédure partitionner(tableau T, entier premier, entier dernier, entier pivot)

Debut

 échanger T[pivot] et T[dernier] ; // échange le pivot avec le dernier du tableau , le pivot devient le dernier du tableau

 j := premier ;

 pour i de premier à dernier - 1 faire// la boucle se termine quand i = (dernier-1).

 si T[i] <= T[dernier] alors

 échanger T[i] et T[j] ;

 j := j + 1 ;

 échanger T[dernier] et T[j] ;

 renvoyer j ;

fin ;

Procédure tri\_rapide(tableau T, entier premier, entier dernier)

 debut

si premier < dernier alors

 pivot := choix\_pivot(T, premier, dernier) ;

 pivot := partitionner(T, premier, dernier, pivot) ;

 tri\_rapide(T, premier, pivot-1) ;

 tri\_rapide(T, pivot+1, dernier) ;

Fin ;