Algorithmique TD 4

Exercice 1

 // 2 instructions

 // 1 instruction

 // 2 instruction

 // 3 instructions

 // 2 instruction

 // 2 instructions

 // 2 instructions

Meilleur cas = Tableau déjà trié en ordre décroissant, on ne rentre jamais dans le tant que.

Pire cas = Tableau trié en ordre croissant, on reste à chaque fois j-1 fois dans le tant que.

*Meilleur cas :*

*Boucle Pour* : 7 instructions répétées (n-1) fois.

*Boucle TantQue* : 3 instruction répétée une (n-1) fois

* 8n-8

*Pire cas :*

*Boucle Pour* : 7 instructions répétées (n-1) fois.

*Boucle TantQue* : 6 instructions répétées une (j-1) fois pour j allant de 2 à n.

* 3n²+4n – 7 (voir cours)

Exercice 2

 *// 1 instruction*

 *// 1 instruction*

 *// 1 instruction*

 *//1 instruction*

 *// 1 instruction*

 *// 1 instruction*

 *// 1 instruction*

 *// 1 instruction*

L’invariant de boucle est le fait que le tableau allant de 1 à j est toujours trié définitivement

On sait que les n-1 premiers éléments sont triés définitivement, donc le dernier élément est nécessairement à la bonne place puisque c’est le plus grand.

Pire cas : tableau trié dans l’ordre décroissant

*Boucle Pour 2* : 3 instructions répétées (taille –(j+1)) pour j allant de 1 à n-1

*Boucle Pour 1* : 5 instructions répétées n-1 fois

Au total : 1,5\*(n-1)\*(n-2) + 5 (n-1) = (n-1)\*(1,5\*(n-2) + 5) = (n-1)\*(1,5n +2)

*Cas moyen :* Une fois sur deux on entre dans le Si => au lieu d’avoir trois instructions on en a 2,5. Ceci ne change pas tellement le résultat.

Exercice 3

**Procedure** triBulle(ES tableau tab(N) : Entier , taille : entier )

 **Variables**

 i,j : entier

 **Pour** j := 1 à taille pas 1

 **Pour** i := 1 à (taille-j)

 **Si** ( tab(i) > tab(i+1) )

 tmp := tab(i)

 tab(i) := tab(i+1)

 tab(i) := tmp

 **Fin Si**

 **Fin Pour**

 **Fin Pour**

**Fin Procedure**

**Procedure** triBulleArret(ES tableau tab(N) : Entier , taille : entier )

 **Variables**

 i,j : entier

 Nontrie : Boolean

 j := 1

 Nontrie := true

 **TantQue** j < taille et Nontrie

 Nontrie = false

 **Pour** i := 1 à (taille-j)

 **Si** ( tab(i) > tab(i+1) )

 tmp := tab(i)

 tab(i) := tab(i+1)

 tab(i) := tmp

 Nontrie := true

 **Fin Si**

 **Fin Pour**

 **i := i+1**

 **Fin Tant Que**

**Fin Procedure**