

DÉPARTEMENT DE MATHÉMATIQUES, D'INFORMATIQUE ET DE GÉNIE

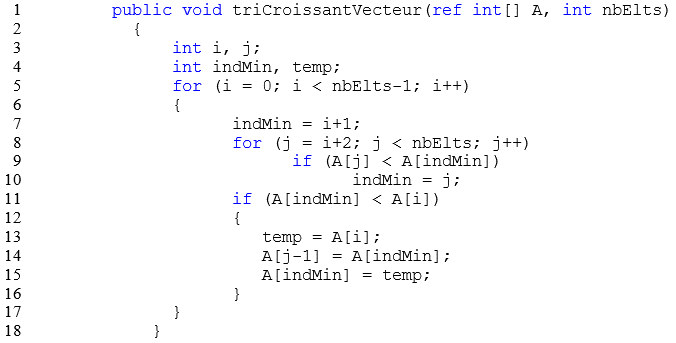
**INF33307 – Assurance de la qualité et gestion**

**de projets informatiques**

**Automne 2023**

**Solution de l’exercice sur les tests**

**Professeur: Ismaïl Khriss**



1. Chemins linéairement indépendants couvrant toutes les instructions :
2. 1-4, 5, 18
3. 1-4, 5, 6-7, 8, 11, 17, 5, 18
4. 1-4, 5, 6-7, 8, 11, 12-16, 17, 5, 18
5. 1-4, 5, 6-7, 8, 9, 10’, 8, 11, 17, 5, 18
6. 1-4, 5, 6-7, 8, 9 10, 10’, 8, 11, 17, 5, 18
7. Donner une suite de tests qui permet de suivre les chemins trouvés

[Fact]

public void TestChemin1()

{

// chemin 2-4, 5, 18

int[] tableau = new int[1] { 0 };

int nombre = 1;

Util.triCroissantVecteur(ref tableau, nombre);

int[] tableauAttendu = new int[1] { 0 };

//XUnit reconnait les collections

Assert.Equal(tableauAttendu, tableau);

}

[Fact]

public void TestChemin2()

{

// chemin 2-4, 5, 6-7, 8, 11, 17, 5, 18

int[] tableau = new int[2] { 0, 1 };

int nombre = 2;

Util.triCroissantVecteur(ref tableau, nombre);

int[] tableauAttendu = new int[2] { 0, 1 };

//XUnit reconnait les collections

Assert.Equal(tableauAttendu, tableau);

}

[Fact]

public void TestChemin3()

{

// chemin 2-4, 5, 6-7, 8, 11, 12-16, 17, 5, 18

int[] tableau = new int[2] { 2, 1 };

int nombre = 2;

Util.triCroissantVecteur(ref tableau, nombre);

int[] tableauAttendu = new int[2] { 1, 2 };

//XUnit reconnait les collections

Assert.Equal(tableauAttendu, tableau);

}

[Fact]

public void TestChemin4()

{

// chemin 2-4, 5, 6-7, 8, 9, 10’, 8, 11, 17, 5, 18

// le test fait plutôt le chemin 2-4, 5, 6-7, 8, 9,

// 10’, 8, 11, 17, 5, 6-7, 8, 11, 17, 5, 18

int[] tableau = new int[3] { 1, 2, 3 };

int nombre = 3;

Util.triCroissantVecteur(ref tableau, nombre);

int[] tableauAttendu = new int[3] { 1, 2, 3 };

//XUnit reconnait les collections

Assert.Equal(tableauAttendu, tableau);

}

[Fact]

public void TestChemin5()

{

// chemin 2-4, 5, 6-7, 8, 9 10, 10’, 8, 11, 17, 5, 18

// le test fait plutôt le chemin 2-4,5,6-7,8,9,10,10',8,11,12-

// 16,17,5,6-7,8,11,12-16,17,5,18

int[] tableau = new int[3] { 2, 3, 1 };

int nombre = 3;

Util.triCroissantVecteur(ref tableau, nombre);

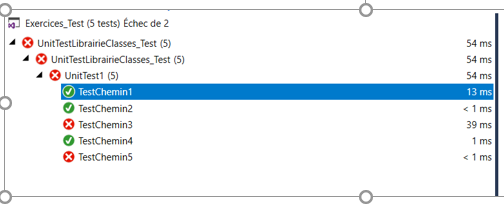
int[] tableauAttendu = new int[3] { 1, 2, 3 };

//XUnit reconnait les collections

Assert.Equal(tableauAttendu, tableau);

}

Chemins 3 et 5 ne fonctionnent pas



Appliquez cette suite de tests sur l’opération. Indiquez pour chaque cas test, si ce dernier réussit ou échoue.

1. Si un de vos cas de test échoue, donnez une version corrigée de l’opération en indiquant le(s) erreur(s) qui s’y trouvai(en)t.

A[j - 1] = A[indMin]; 🡺 A[i] = A[indMin];

1. Faites les tests de régression sur la nouvelle version de l’algorithme.

