Département de génie informatique et de génie logiciel

LOG3430
Méthodes de tests et validations

TP2
Tests combinatoires et couverture de code

Soumis par :
Jean-Frédéric Fontaine (1856632)
Sébastien Cadorette (1734603)

15 octobre 2018
Soumis à : Hiba Bagane

Polytechnique Montréal

Tests EC

Tout d'abord, on dresse la liste des choix et contraintes, question de construire les spécifications formelles des tests. Une fois les spécifications déterminées, on élabore une trame de tests en respectant les contraintes. Pour cette section, on utilise le critère EC (*Each Choice*), c'est-à-dire que pour notre trame de tests, on sélectionne un choix de chaque catégorie et on leur donne une valeur. La liste complète des spécifications formelles et des trames de tests est dans la section des tests AC (*All Combinations*). On retrouve ici les choix sélectionnés pour répondre à la contrainte EC.

Méthodes

Méthode simple (int V, int E)

Trame de tests

V1E4 -> a4 = <{ v = -4 , e = 20 } , { erreur }>
V2E3 -> a7 = <{ v = 0 , e = 0 } , { grapheOK }>
V3E1 -> a9 = <{ v = 4 , e = -1 } , { erreur }>
V3E2 -> a10 = <{ v = 4 , e = 0 } , { grapheOK }>

Méthode simple (int V, double P)

Trame de tests

V1P5 -> b5 = <{ v = -4 , p = 1.5 } , { erreur }>
V2P1 -> b6 = <{ v = 0 , p = -1.0 } , { erreur }>
V2P2 -> b7 = <{ v = 0 , p = 0.0 } , { grapheOK }>
V3P3 -> b13 = <{ v = 10 , p = 0.5 } , { grapheOK }>
V3P4 -> b14 = <{ v = 10 , p = 1.0 } , { grapheOK }>

Méthode bipartie (int V1, int V2, int E)

Trame de tests

V11V21E4 -> c4 = <{ v1 = -4 , v2 = -4 , e = 20 } , { erreur }>
V12V22E1 -> c17 = <{ v1 = 0 , v2 = 0 , e = -4 } , { erreur }>
V13V22E3 -> c31 = <{ v1 = 4 , v2 = 0 , e = 0 } , { grapheOK }>
V13V23E2 -> c34 = <{ v1 = 4 , v2 = 4 , e = 0 } , { grapheOK }>

Méthode bipartie (int V1, int V2, double P)

Trame de tests

V11V22P5 -> d10 = <{ v1 = -4 , v2 = 0 , p = 1.5 } , { erreur }>
V12V23P4 -> d29 = <{ v1 = 0 , v2 = 4 , p = 1.0 } , { grapheOK }>
V13V21P1 -> d31 = <{ v1 = 4 , v2 = -4 , p = -1.0 } , { erreur }>
V13V23P2 -> d42 = <{ v1 = 4 , v2 = 4 , p = 0.0 } , { grapheOK }>
V13V23P3 -> d43 = <{ v1 = 4 , v2 = 4 , p = 0.5 } , { grapheOK }>

Méthode regular (int V, int K)

Trame de tests

V1K1 -> e1 = <{ v = -1, k = -1 } , { erreur }>
V2K3 -> e6 = <{ v = 4, k = 5 } , { grapheOK }>
V3K2 -> e8 = <{ v = 5, k = 2 } , { grapheOK }>

Couverture

Une fois tous les tests écrient et fonctionnelles, on calcule la couverture de nos tests.

Tout d'abord, la méthode simple (int V, int E) est couverte à 60,3%. Cela est dû au fait que lors de la sélection de nos tests en respectant le critère EC, nos tests ne lançant pas d'erreurs ont tous 0 comme valeur pour les arrêtes. Cela a pour effet qu'une partie de la fonction n'est pas exécutée en raison d'une condition de boucle. Bref, après l'ajout d'un test avec un nombre supérieur à 0 pour les arrêtes, on obtient maintenant une couverture de 100%.
Par la suite, la méthode simple (int V, double P) est couverte à 100%. Pas d'ajustement à lui apporter.
Ensuite, la méthode bipartite (int V1, int V2, int E) est couverte à 63,4%. La raison est la même que pour la méthode simple (int V, int E). Avec l'ajout d'un test contenant un nombre d'arrêtes supérieur à 0, on calcule une couverture de 100%.
Par la suite, la méthode bipartite (int V1, int V2, double P) est couverte à 100%. Pas d'ajustement à lui apporter.
Pour terminer, la méthode regular (int V, int K) est couverte à 100%. Pas d'ajustement à lui apporter.

Tests AC

Pour le critère AC (*All Combinations*), chaque choix doit être combiné à chacun des autres choix des catégories. Un se retrouve ainsi avec une liste bien plus exhaustive que pour le critère EC. Il faut également respecter les contraintes des spécifications formelles. Les tests ne respectant pas les contraintes sont marqués en rouge. Ils ne seront pas utilisés dans le programme Java.

Méthodes

Méthode simple (int V, int E)

Spécifications formelles

V1 =	V <= -1	[propriété : erreur]
V2 =	V = 0	[propriété : grapheOK, grapheVide]
V3 =	V >= 1	[propriété : grapheOK]
E1 =	E <= -1	[propriété : erreur]
E2 =	E = 0	[si grapheOK]
E3 =	0 < E < Vx(V-1)/2	[si grapheOK]
E4 =	E >= Vx(V-1)/2	[propriété : erreur]

Trame de tests

V1E1 -> a1 =	<{ v = -4 , e = -1 }	, { erreur }>
V1E2 -> a2 =	<{ v = -4 , e = 0 }	, { erreur }>
V1E3 -> a3 =	<{ v = -4 , e = 4 }	, { erreur }>
V1E4 -> a4 =	<{ v = -4 , e = 20 }	, { erreur }>
V2E1 -> a5 =	<{ v = 0 , e = -1 }	, { erreur }>
V2E2 -> a6 =	<{ v = 0 , e = 0 }	, { grapheOK }>
V2E3 -> a7 =	<{ v = 0 , e = 0 }	, { grapheOK }>
V2E4 -> a8 =	<{ v = 0 , e = 20 }	, { erreur }>
V3E1 -> a9 =	<{ v = 4 , e = -1 }	, { erreur }>
V3E2 -> a10 =	<{ v = 4 , e = 0 }	, { grapheOK }>
V3E3 -> a11 =	<{ v = 4 , e = 4 }	, { grapheOK }>

V3E4 -> a12 = <{ v = 4 , e = 20 } , { erreur }>

Méthode simple (int V, double P)

Spécifications formelles

V1 = V <= -1 [propriété : erreur]
V2 = V = 0 [propriété : grapheOK, grapheVide]
V3 = V >= 1 [propriété : grapheOK]
P1 = P < 0.0 [propriété : erreur]
P2 = P = 0.0 [si grapheOK]
P3 = 0.0 < P < 1.0 [si grapheOK]
P4 = P = 1.0 [si grapheOK]
P5 = P > 1.0 [propriété : erreur]

Trame de tests

V1P1 -> b1 = <{ v = -4 , p = -1.0 } , { erreur }>
V1P2 -> b2 = <{ v = -4 , p = 0.0 } , { erreur }>
V1P3 -> b3 = <{ v = -4 , p = 0.5 } , { erreur }>
V1P4 -> b4 = <{ v = -4 , p = 1.0 } , { erreur }>
V1P5 -> b5 = <{ v = -4 , p = 1.5 } , { erreur }>
V2P1 -> b6 = <{ v = 0 , p = -1.0 } , { erreur }>
V2P2 -> b7 = <{ v = 0 , p = 0.0 } , { grapheOK }>
V2P3 -> b8 = <{ v = 0 , p = 0.5 } , { grapheOK }>
V2P4 -> b9 = <{ v = 0 , p = 1.0 } , { grapheOK }>
V2P5 -> b10 = <{ v = 0 , p = 1.5 } , { erreur }>
V3P1 -> b11 = <{ v = 10 , p = -1.0 } , { erreur }>
V3P2 -> b12 = <{ v = 10 , p = 0.0 } , { grapheOK }>
V3P3 -> b13 = <{ v = 10 , p = 0.5 } , { grapheOK }>
V3P4 -> b14 = <{ v = 10 , p = 1.0 } , { grapheOK }>
V3P5 -> b15 = <{ v = 10 , p = 1.5 } , { erreur }>

Méthode bipartie (int V1, int V2, int E)

Spécifications formelles

V11 = V <= -1 [propriété : erreur]
V12 = V = 0 [propriété : vertice1OK, vertice1Vide]
V13 = V >= 1 [propriété : vertice1OK]
V21 = V <= -1 [propriété : erreur]
V22 = V = 0 [propriété : vertice2OK, vertice2Vide]
V23 = V >= 1 [propriété : vertice2OK]
E1 = E < 0 [propriété : erreur]
E2 = E = 0 [si vertice1OK et vertice2OK]
E3 = 0 < E <= V1*V2 [si vertice1OK et vertice2OK]
E4 = E > V1*V2 [propriété : erreur]

Trame de tests

V11V21E1 -> c1 = <{ v1 = -4 , v2 = -4 , e = -4 } , { erreur }>
V11V21E2 -> c2 = <{ v1 = -4 , v2 = -4 , e = 0 } , { erreur }>
V11V21E3 -> c3 = <{ v1 = -4 , v2 = -4 , e = 12 } , { erreur }>
V11V21E4 -> c4 = <{ v1 = -4 , v2 = -4 , e = 20 } , { erreur }>
V11V22E1 -> c5 = <{ v1 = -4 , v2 = 0 , e = -4 } , { erreur }>

V11V22E2 -> c6 = <{ v1 = -4 , v2 = 0 , e = 0 } , { erreur }>
V11V22E3 -> c7 = <{ v1 = -4 , v2 = 0 , e = 0 } , { erreur }>
V11V22E4 -> c8 = <{ v1 = -4 , v2 = 0 , e = 20 } , { erreur }>
V11V23E1 -> c9 = <{ v1 = -4 , v2 = 4 , e = -4 } , { erreur }>
V11V23E2 -> c10 = <{ v1 = -4 , v2 = 4 , e = 0 } , { erreur }>
V11V23E3 -> c11 = <{ v1 = -4 , v2 = 4 , e = 0 } , { erreur }>
V11V23E4 -> c12 = <{ v1 = -4 , v2 = 4 , e = 20 } , { erreur }>
V12V21E1 -> c13 = <{ v1 = 0 , v2 = -4 , e = -4 } , { erreur }>
V12V21E2 -> c14 = <{ v1 = 0 , v2 = -4 , e = 0 } , { erreur }>
V12V21E3 -> c15 = <{ v1 = 0 , v2 = -4 , e = 0 } , { erreur }>
V12V21E4 -> c16 = <{ v1 = 0 , v2 = -4 , e = 20 } , { erreur }>
V12V22E1 -> c17 = <{ v1 = 0 , v2 = 0 , e = -4 } , { erreur }>
V12V22E2 -> c18 = <{ v1 = 0 , v2 = 0 , e = 0 } , { grapheOK }>
V12V22E3 -> c19 = <{ v1 = 0 , v2 = 0 , e = 0 } , { grapheOK }>
V12V22E4 -> c20 = <{ v1 = 0 , v2 = 0 , e = 20 } , { erreur }>
V12V23E1 -> c21 = <{ v1 = 0 , v2 = 4 , e = -4 } , { erreur }>
V12V23E2 -> c22 = <{ v1 = 0 , v2 = 4 , e = 0 } , { grapheOK }>
V12V23E3 -> c23 = <{ v1 = 0 , v2 = 4 , e = 0 } , { grapheOK }>
V12V23E4 -> c24 = <{ v1 = 0 , v2 = 4 , e = 20 } , { erreur }>
V13V21E1 -> c25 = <{ v1 = 4 , v2 = -4 , e = -4 } , { erreur }>
V13V21E2 -> c26 = <{ v1 = 4 , v2 = -4 , e = 0 } , { erreur }>
V13V21E3 -> c27 = <{ v1 = 4 , v2 = -4 , e = 0 } , { erreur }>
V13V21E4 -> c28 = <{ v1 = 4 , v2 = -4 , e = 20 } , { erreur }>
V13V22E1 -> c29 = <{ v1 = 4 , v2 = 0 , e = -4 } , { erreur }>
V13V22E2 -> c30 = <{ v1 = 4 , v2 = 0 , e = 0 } , { grapheOK }>
V13V22E3 -> c31 = <{ v1 = 4 , v2 = 0 , e = 0 } , { grapheOK }>
V13V22E4 -> c32 = <{ v1 = 4 , v2 = 0 , e = 20 } , { erreur }>
V13V23E1 -> c33 = <{ v1 = 4 , v2 = 4 , e = -4 } , { erreur }>
V13V23E2 -> c34 = <{ v1 = 4 , v2 = 4 , e = 0 } , { grapheOK }>
V13V23E3 -> c35 = <{ v1 = 4 , v2 = 4 , e = 12 } , { grapheOK }>
V13V23E4 -> c36 = <{ v1 = 4 , v2 = 4 , e = 20 } , { erreur }>

Méthode bipartie (int V1, int V2, double P)

Spécifications formelles

V11 = V <= -1 [propriété : erreur]
V12 = V = 0 [propriété : vertice1OK, vertice1Vide]
V13 = V >= 1 [propriété : vertice1OK]
V21 = V <= -1 [propriété : erreur]
V22 = V = 0 [propriété : vertice2OK, vertice2Vide]
V23 = V >= 1 [propriété : vertice2OK]
P1 = P < 0.0 [propriété : erreur]
P2 = P = 0.0 [si vertice1OK et vertice2OK]
P3 = 0.0 < P < 1.0 [si vertice1OK et vertice2OK]
P4 = P = 1.0 [si vertice1OK et vertice2OK]
P5 = P > 1.0 [propriété : erreur]

Trame de tests

V11V21P1 -> d1 = <{ v1 = -4 , v2 = -4 , p = -1.0 } , { erreur }>

V11V21P2 -> d2 = <{ v1 = -4 , v2 = -4 , p = 0.0 } , { erreur }>
V11V21P3 -> d3 = <{ v1 = -4 , v2 = -4 , p = 0.5 } , { erreur }>
V11V21P4 -> d4 = <{ v1 = -4 , v2 = -4 , p = 1.0 } , { erreur }>
V11V21P5 -> d5 = <{ v1 = -4 , v2 = -4 , p = 1.5 } , { erreur }>
V11V22P1 -> d6 = <{ v1 = -4 , v2 = 0 , p = -1.0 } , { erreur }>
V11V22P2 -> d7 = <{ v1 = -4 , v2 = 0 , p = 0.0 } , { erreur }>
V11V22P3 -> d8 = <{ v1 = -4 , v2 = 0 , p = 0.5 } , { erreur }>
V11V22P4 -> d9 = <{ v1 = -4 , v2 = 0 , p = 1.0 } , { erreur }>
V11V22P5 -> d10 = <{ v1 = -4 , v2 = 0 , p = 1.5 } , { erreur }>
V11V23P1 -> d11 = <{ v1 = -4 , v2 = 4 , p = -1.0 } , { erreur }>
V11V23P2 -> d12 = <{ v1 = -4 , v2 = 4 , p = 0.0 } , { erreur }>
V11V23P3 -> d13 = <{ v1 = -4 , v2 = 4 , p = 0.5 } , { erreur }>
V11V23P4 -> d14 = <{ v1 = -4 , v2 = 4 , p = 1.0 } , { erreur }>
V11V23P5 -> d15 = <{ v1 = -4 , v2 = 4 , p = 1.5 } , { erreur }>
V12V21P1 -> d16 = <{ v1 = 0 , v2 = -4 , p = -1.0 } , { erreur }>
V12V21P2 -> d17 = <{ v1 = 0 , v2 = -4 , p = 0.0 } , { erreur }>
V12V21P3 -> d18 = <{ v1 = 0 , v2 = -4 , p = 0.5 } , { erreur }>
V12V21P4 -> d19 = <{ v1 = 0 , v2 = -4 , p = 1.0 } , { erreur }>
V12V21P5 -> d20 = <{ v1 = 0 , v2 = -4 , p = 1.5 } , { erreur }>
V12V22P1 -> d21 = <{ v1 = 0 , v2 = 0 , p = -1.0 } , { erreur }>
V12V22P2 -> d22 = <{ v1 = 0 , v2 = 0 , p = 0.0 } , { grapheOK }>
V12V22P3 -> d23 = <{ v1 = 0 , v2 = 0 , p = 0.5 } , { grapheOK }>
V12V22P4 -> d24 = <{ v1 = 0 , v2 = 0 , p = 1.0 } , { grapheOK }>
V12V22P5 -> d25 = <{ v1 = 0 , v2 = 0 , p = 1.5 } , { erreur }>
V12V23P1 -> d26 = <{ v1 = 0 , v2 = 4 , p = -1.0 } , { erreur }>
V12V23P2 -> d27 = <{ v1 = 0 , v2 = 4 , p = 0.0 } , { grapheOK }>
V12V23P3 -> d28 = <{ v1 = 0 , v2 = 4 , p = 0.5 } , { grapheOK }>
V12V23P4 -> d29 = <{ v1 = 0 , v2 = 4 , p = 1.0 } , { grapheOK }>
V12V23P5 -> d30 = <{ v1 = 0 , v2 = 4 , p = 1.5 } , { erreur }>
V13V21P1 -> d31 = <{ v1 = 4 , v2 = -4 , p = -1.0 } , { erreur }>
V13V21P2 -> d32 = <{ v1 = 4 , v2 = -4 , p = 0.0 } , { erreur }>
V13V21P3 -> d33 = <{ v1 = 4 , v2 = -4 , p = 0.5 } , { erreur }>
V13V21P4 -> d34 = <{ v1 = 4 , v2 = -4 , p = 1.0 } , { erreur }>
V13V21P5 -> d35 = <{ v1 = 4 , v2 = -4 , p = 1.5 } , { erreur }>
V13V22P1 -> d36 = <{ v1 = 4 , v2 = 0 , p = -1.0 } , { erreur }>
V13V22P2 -> d37 = <{ v1 = 4 , v2 = 0 , p = 0.0 } , { grapheOK }>
V13V22P3 -> d38 = <{ v1 = 4 , v2 = 0 , p = 0.5 } , { grapheOK }>
V13V22P4 -> d39 = <{ v1 = 4 , v2 = 0 , p = 1.0 } , { grapheOK }>
V13V22P5 -> d40 = <{ v1 = 4 , v2 = 0 , p = 1.5 } , { erreur }>
V13V23P1 -> d41 = <{ v1 = 4 , v2 = 4 , p = -1.0 } , { erreur }>
V13V23P2 -> d42 = <{ v1 = 4 , v2 = 4 , p = 0.0 } , { grapheOK }>
V13V23P3 -> d43 = <{ v1 = 4 , v2 = 4 , p = 0.5 } , { grapheOK }>
V13V23P4 -> d44 = <{ v1 = 4 , v2 = 4 , p = 1.0 } , { grapheOK }>
V13V23P4 -> d45 = <{ v1 = 4 , v2 = 4 , p = 1.5 } , { erreur }>

Méthode regular (int V, int K)

Spécifications formelles

V1 = V < 0 [propriété : erreur]
V2 = V % 2 = 0 [propriété : vOK & vPair]
V3 = V % 2 = 1 [propriété : vOK & vImpair]
K1 = K < 0 [propriété : erreur]
K2 = K % 2 = 0 [si vOK]
K3 = K % 2 = 1 [si vOK & vPair]

Trame de tests

V1K1 -> e1 = <{ v = -1, k = -1 } , { erreur }>
V1K2 -> e2 = <{ v = -1, k = 4 } , { erreur }>
V1K3 -> e3 = <{ v = -1, k = 5 } , { erreur }>
V2K1 -> e4 = <{ v = 4, k = -1 } , { erreur }>
V2K2 -> e5 = <{ v = 4, k = 2 } , { grapheOK }>
V2K3 -> e6 = <{ v = 4, k = 5 } , { grapheOK }>
V3K1 -> e7 = <{ v = 5, k = -1 } , { erreur }>
V3K2 -> e8 = <{ v = 5, k = 2 } , { grapheOK }>
V3K3 -> e9 = <{ v = 5, k = 5 } , { erreur }>