

**Travail pratique #2 - Tests combinatoires et couverture de code**  
**Automne 2018**

### **1. Mise en contexte théorique**

Tels que présenté dans le cours, il existe trois critères d'adéquation basés sur le choix qu'on peut utiliser pour réaliser des tests en boîte noire, appelés tests combinatoires. Premièrement, on trouve le critère **AC** (*All Choice*) qui est le critère est le plus utilisé. Selon ce critère, chaque choix doit être utilisé avec tous les choix des autres catégories. Ensuite, nous avons **EC** (*Each choice*) et selon ce critère, chaque valeur de chaque choix pour chaque catégorie doit être utilisée dans un cas de test. Finalement, on trouve le critère **BC** (*Basic Choice*), qui représente un compromis entre les critères **AC** et **EC**. Dans le cadre de ce laboratoire, nous allons utiliser seulement les critères **AC** et **EC**. Pour plus d'explications et des exemples, voir les notes de cours sur le chapitre catégorie-partition.

### **2. Objectifs**

- Comprendre et mettre en pratiques les notions théoriques vues dans le cours.
- Se familiariser avec le framework JUnit et l'outil JaCoCo

### **3. Mise en contexte pratique**

Le projet de ce laboratoire regroupe les algorithmes et les structures de données les plus populaires écrits en Java. Pour ce travail, vous allez tester seulement la classe `GraphGenerator`. Il est nécessaire d'avoir des connaissances de base en Java et en théorie des graphes pour réaliser ce TP. La théorie des graphes a été enseigné dans le cours LOG2810 - Structures discrètes et le langage Java dans le cours INF2010 - Structures de données et algorithmes.

La première partie du travail à effectuer se réalise en boîte noire, donc vous ne vous basez pas sur l'implémentation des méthodes. Vous allez vous référez aux paramètres de ces méthodes et vos connaissances sur les graphes.

La deuxième partie se réalise en boîte blanche, donc vous devez comprendre l'implémentation de la méthode pour réaliser vos cas de tests.

## 4. Travail à effectuer

### 4.1. Méthodes combinatoires

Vous devez tester les méthodes de génération des graphes simples, bipartis et réguliers de la classe `GraphGenerator` en utilisant la technique de test boîte noire catégorie-partition **EC** (*Each Choice*).

Les méthodes sont les suivantes :

- **Graph** simple(**int** V, **int** E)
- **Graph** simple(**int** V, **double** p)
- **Graph** bipartite(**int** V1, **int** V2, **int** E)
- **Graph** bipartite(**int** V1, **int** V2, **double** p)
- **Graph** regular(**int** V, **int** k)

Chaque méthode de test devra être commentée en identifiant le cas de test en question.

Refaites le même travail demandé, en utilisant le critère **AC** (*All Combinations*). Chaque méthode de test correspond à un cas de test boîte blanche. La méthode de test devra être commentée en identifiant le cas de test en question.

### 4.2. Couverture des méthodes

À l'aide de l'outil JaCoCo, calculez la couverture des méthodes testées avec le critère **EC** (*Each Choice*). Si vous n'obtenez pas une couverture de 100%, expliquez pourquoi. Ensuite, corrigez vos cas de test (dans un fichier séparé) afin d'obtenir une couverture complète. Sinon s'il est impossible d'avoir une couverture de 100%, expliquez pourquoi.

## 5. Questions

1. Faites un retour sur le TP précédent (commentaires, difficultés rencontrées etc).
2. Expliquez comment vous avez obtenu vos cas de tests pour la partie **4.1**.

## 6. Livrable à remettre, procédure de remise et retard

1. Vos fichiers de test \*.java. **NB: votre package doit être nommé TP2**
2. Le rapport JaCoCo au format HTML pour chacune des stratégies :
  - a. Un rapport intermédiaire à la fin des tests boîte noire
  - b. Un rapport final, une fois les tests boîte blanche implémentés et exécutés
3. Un rapport de cinq pages au maximum (PDF) contenant les réponses aux questions.

Veuillez envoyer vos fichiers dans une archive de type \*.zip (et seulement zip, pas de rar, 7z, etc) qui portera le nom : **LOG3430\_lab2\_MatriculeX\_MatriculeY.zip** tel que :  
MatriculeX < MatriculeY.

**Date limite pour la remise :**

**Groupe B1 & B2 : Lundi 15 octobre à 7:00 du matin**

## **7. Barème de correction**

### **Rapport de 4 pages**

**1.5 points**

- Pertinence des jeux de tests
- Pertinence des explications et des analyses
- Réponses aux questions
- Respect du nombre de pages

### **Tests**

**3.5 points**

- Respect des consignes
- Tests exécutables
- Qualité du code

Les travaux en retard seront pénalisés de 20% par jour de retard. Aucun travail ne sera accepté après 4 jours de retard. Si votre dépôt ne respecte pas la nomenclature définie ci-dessus, 0.5 point de pénalité sera appliqué.

## **8. Documentation**

Consultez le complément au TP2 sur moodle.