

# LOG3430 - MÉTHODES DE TEST ET DE VALIDATION DU LOGICIEL

---

## LABORATOIRE 2

### TESTS COMBINATOIRES ET COUVERTURE DE CODE

Département de génie informatique et de génie logiciel  
École Polytechnique de Montréal



Automne 2019

## 1 Mise en contexte théorique

Tel que présenté dans le cours, il existe trois critères d'adéquation basés sur le choix que nous pouvons utiliser pour réaliser des tests en boîte noire, appelés tests combinatoires.

Premièrement, nous trouvons le critère **AC (All Choice)** qui 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, nous trouvons 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 unittest et l'outil Coverage.

Consultez le lien suivant :

<https://coverage.readthedocs.io/en/v4.5.x/>

## 3 Mise en contexte pratique

Pour ce laboratoire ,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 théorie des graphes a été enseigné dans le cours LOG2810 - Structures discrètes. 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

Vous devez tester les méthodes de génération des graphes du fichier *generators.py* en utilisant la technique de test boîte noire **catégorie-partition EC (Each Choice)**. Les méthodes sont les suivantes :

- `simple(int V, int E)`
- `simple_with_probability (int V, double p)`
- `bipartite (int V1, int V2, int E)`
- `bipartite_with_probability (int V1,int V2, double p)`
- `eurlerianCycle (int V, int E)`
- `regular ( int V, int k)`

Chaque méthode de test devra être commentée en identifiant le cas de test en question.**4.5 points**

Refaites le même travail demandé, en utilisant le critère **AC (All Combinations)**. Chaque méthode de test correspond à un cas de test. La méthode de test devra être commentée en identifiant le cas de test en question.**4.5 points**

## Couverture des méthodes et tests en boîte blanche

À l'aide de l'outil Coverage, calculez la couverture des méthodes testées avec le critère **EC(Each Choice)**. Si vous n'obtenez pas une couverture de 100%, expliquez pourquoi.**4.5 points** Construisez les graphes de flot de contrôle **CFG** pour les méthodes testées précédemment. Ensuite, identifiez tous les chemins à tester et implémentez ceux qui ne sont pas couverts par vos tests **EC.4.5 points**

## 5 Question

Est-ce que le code est conforme à sa spécification( commentaires) ? Sinon faites les changements adéquates.**2 point**

## 6 Livrables attendus

Les livrables suivants sont attendus :

- Un rapport pour le laboratoire.
- Le dossier COMPLET contenant le projet.

Le tout à remettre dans une seule archive **zip** avec pour nom matricule1\_matricule2\_lab1.zip à téléverser sur Moodle.

Le rapport doit contenir le titre et numéro du laboratoire, les noms et matricules des coéquipiers ainsi que le numéro du groupe.

**Consultez le site Moodle du cours pour la date et l'heure limites de remise des fichiers.** Un retard de ]0,24h] sera pénalisé de 10%, de ]24h, 48h] de 20% et de plus de 48h de 50%.