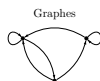


# IFT2015 - Automne 2009

## Structures de données



Sylvie Hamel

André-Aisenstadt: 3161  
hamelsyl@iro.umontreal.ca

<http://www.iro.umontreal.ca/~hamelsyl/IFT2015-A09>

## Préalables:

- IFT 1025: Programmation 2
- IFT 1065: Structures discrètes en informatique

## Horaire et locaux:

<i>Activité</i>	<i>Jour</i>	<i>Local</i>
théorie	Lundi: 10h30-12h30	1177 A.A.
théorie	Mardi: 10h30-11h30	1140 A.A.
travaux pratiques	Vendredi: 11h30-13h30	1355 A.A.

## Objectifs:

Le cours IFT2015 vise à familiariser l'étudiant avec les techniques de base pour l'organisation, la manipulation et la recherche de données. Le but du cours est d'être capable de choisir les structures de données qui rendent un programme le plus efficace possible, que ce soit en terme du temps d'exécution ou de l'espace mémoire utilisé.

## Manuel obligatoire:

Robert Sedgewick, **Algorithms en Java, Parts 1-4, 3rd edition**, Pearson Education, 2003.

- Livre orienté pour les programmeurs JAVA
- Livre est complet et facile à lire
- Disponible à la librairie de l'université
- Disponible aussi en français

Robert Sedgewick, **Algorithmes en Java, 3ieme edition**, Pearson Education, 2004.

## Contenu:

1. Introduction et rappels: Mesure de complexité, types abstraits de données, listes chaînées, piles, files. (Chapitres 1-4 approximativement 6 heures)
2. Files avec priorité. (Chapitre 5,9, approximativement 3 heures)
3. Arbres: Rappels sur les arbres binaires de recherche, arbres AVL, arbres de recherche généralisés, arbres (2,4), arbres rouges-noirs. (Chapitres 5 et 13, approximativement 12 heures)
4. Dictionnaires: Tables et fonctions de hachage, méthode de recherche avec mémoire externe. (Chapitres 14, approximativement 6 heures)
5. Graphes: graphes orientés, graphes avec poids, parcours de graphes, arbre minimal couvrant un graphe, récupération de mémoire. (Notes de cours, approximativement 12 heures)

## Manuels de références:

1. Goodrich M.T., Tamassia R., **Data structures and algorithms in Java (4th edition)**, John Wiley et Sons, Inc., 2006.
2. Mark Allen Weiss, **Data Structures and Algorithm Analysis in Java (2nd edition)**, Pearson Education, 2007.
3. T.A. Standish, **Data Structures in JAVA**, Addison Wesley, 1998.
4. A.Aho, J.E. Hopcroft and J.D. Ullman, **Data Structures and Algorithms**, Addison Wesley, 1983.
5. D.E. Knuth, **The Art of Computer Programming (Second Edition)**, Addison Wesley, 1973.

## Journées Spéciales:

- Premier TP: **Vendredi 11 septembre**, 11h30-12h30
- Intra: **Vendredi 16 octobre**, 11h30-13h30, AA1355
- Final: **Lundi 14 décembre**, 9h30-12h30, S-142
- Semaine de lecture: **19 au 23 octobre**

## Démonstrateur:

- À déterminer
- Disponibilité: À déterminer

## Évaluation: ▲ 2 devoirs:

- Pondération: 15 % chacun
- Discussions permises
- TP1: - réception de l'énoncé: **22 septembre**  
- remise du devoir: **6 octobre**
- TP2: - réception de l'énoncé: **10 novembre**  
- remise du devoir: **24 novembre**

Tout retard dans la remise des travaux entraînera une pénalité de 10% par jour (24 heures).

### ▲ Intra(Vendredi 16 octobre):

- Pondération: 30 %
- Durée: 2 heures

### ▲ Final(Lundi 14 décembre):

- Pondération: 40 %
- Durée: 3 heures

} Un seuil cumulatif de 50% aux examens sera appliqué (moyenne de 50% pour les deux examens combinés)

## Plagiat:

- Pour vos travaux, vous pouvez utiliser tout ce qui est disponible et accessible publiquement en autant que les références utilisées soient proprement citées.
- Je considère que tout étudiant à lu, signé et respectera le code d'honneur du DIRO
- Les cas de plagiat seront traités conformément aux règles en vigueur à l'Université de Montréal.

## Qu'est-ce qu'une structure de données:

Le terme structure de données, représente à la fois:

- un moyen pour organiser les données en mémoire
- des méthodes (algorithmes) pour manipuler ces données
  - ajout et suppression efficace de données
  - accès aux données, etc...

## Pourquoi étudier les structures de données?

- tous les langages de programmation utilise le même genre de structures de données
- le bon choix de structures peut faire augmenter l'efficacité des algorithmes implémentés que ce soit au niveau de l'efficacité en temps ou de l'efficacité en espace mémoire
- d'un point de vue théorique, beaucoup de structures utilisent des algorithmes très intéressants pour l'accès, l'ajout et la suppression de données

## Comment choisir la bonne structure?

Nous verrons pendant ce cours beaucoup de structures de données différentes

- tableau
- liste chaînée
- pile
- file
- tas ou monceau
- arbres de recherche
- table de hachage
- graphes, etc.

Le choix de la "meilleure" structure de données dans un cas particulier dépendra en grande partie des opérations sur les données les plus utilisées dans ce cas.

Par exemple, si l'accès aux données est très importante mais l'ajout et le retrait beaucoup moins, nous choisirons une structure permettant un accès très efficaces aux données, même si cette structure est moins efficace pour l'ajout et le retrait de données.