

IFT2015 STRUCTURES DE DONNÉES  
LISTE DÉTAILLÉE DE SUJETS À L'INTRA

Miklós Csűrös

Département d'informatique et de recherche opérationnelle  
Université de Montréal

Automne 2016

## **F0 Introduction**

Le but de ce document est de définir les connaissances requises dans le cours IFT2015 à l'examen intra. Cet examen constitue aussi la première partie de l'examen pré-doctorale en structures de données.

- ◆ La connaissance des sujets marqués par \* est exigée pour un «B/A-». Les sujets marqués par \*\* correspondent plutôt à un niveau «A+/A».
- ★ Les notes marginales sont des références aux ouvrages suivants
  - S** Sedgewick, R. *Algorithmes en Java*, 3<sup>e</sup> édition (2004)
  - SW** Sedgewick, R. et K. Wayne. *Algorithms*, 4<sup>e</sup> édition (2011)
- ★ Les notes de cours et des liens vers des articles Wikipedia sont affichés sur le site <http://ift2015a16.wordpress.com/>.
- ★ Aucune documentation ne sera permise à l'examen intra.

# F1 Principes d'analyse d'algorithmes

## Références

- ▷ Sedgewick chapitre 2; Sedgewick & Wayne section §1.4
- ▷ Note sur les fondations : `notes01-recursion.pdf`.
- ▷ Note sur l'analyse d'algorithmes : `notes04-analysis.pdf`.

## Sujets

- ★ Principes de base : pire cas, meilleur cas, moyen cas. S§2.1,2.2,2.7
- ★ Croissance de fonctions communes : constantes, logarithmiques, polynomiales, exponentielles. Factorielle ( $n!$ ), approximation de Stirling, nombres Fibonacci ( $F_n = F_{n-1} + F_{n-2}$ ), nombres harmoniques ( $H_n = \sum_{i=1}^n 1/i = \ln n + \gamma + o(1)$ ). S§2.3
- ★ Notion de temps amorti.
- ★★ Preuves de résultats sur le coût amorti d'opérations. Principe d'analyse crédit/débit.
- ★ Notation asymptotique : définitions de grand  $O(f)$ , petit  $o(f)$ ,  $\Theta(f)$  et  $\Omega(f)$ . S§2.4  
Asymptotiques exactes  $f \sim g$ . Expressions avec  $O()$  ou  $o()$ , règles d'arithmétique :  $O(f) + O(g)$ ,  $O(f) \cdot O(g)$ . Relations avec la limite

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c > 0 \quad \Rightarrow \quad f(n) = O(g(n));$$

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0 \quad \Leftrightarrow \quad f(n) = o(g(n));$$

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1 \quad \Leftrightarrow \quad f(n) \sim g(n)$$

- ★ Application directe de la définition pour démontrer  $f = O(g)$  ou  $f = o(g)$ .
- ★★ Preuve par induction pour récurrences asymptotiques.
- ★ Détermination informelle du temps de calcul et d'usage de mémoire pour algorithmes (itératifs) simples
- ★ Récurrences simples. S§2.5,2.6

$$\begin{array}{ll} f(n) = f(n-1) + O(1) & f(n) = O(n); \\ f(n) = f(n-1) + O(n) & f(n) = O(n^2); \\ f(n) = f(n/2) + O(1) & f(n) = O(\log n); \\ f(n) = f(n/2) + O(n) & f(n) = O(n); \end{array}$$

- ★ Validation expérimentale de temps de calcul.

## F2 Structures élémentaires et types abstraits

### Références

- ▷ Sedgewick chapitres 3 et 4 ; Sedgewick & Wayne sections 1.1–1.3
- ▷ Note sur les listes : `notes02-linkedlist.pdf`.
- ▷ Note sur les tableaux : `notes03-tableaux.pdf`.

### Sujets

- ★ Blocs de construction pour programmes Java. S§3.1 ; SW§1.1
- ★ Tableaux. S§3.2
- ★ Listes chaînées. Variations : listes circulaires, doublement chaînées. Sentinelles pour la tête et/ou la queue. Manipulation d'éléments sur la liste, insertion et suppression. Parcours d'une liste. S§3.3,3.4
- ★ Gestion de mémoire pour listes. S§3.5
- ★ Notion d'un type abstrait, interface, implantation, client. S§4.1 ; SW§1.2
- ★ Types abstraits de files généralisées, piles et queues/files FIFO. S§4.2,4.7
- ★ Implantations de pile et de queue par tableaux ou listes chaînées. Efficacité d'implantations différentes (temps de calcul pour les opérations standardes). Débordement. S§4.4,4.5,4.7 ; SW§1.3



◀ français

English ▶

## E0 Introduction

This document defines the skills and knowledge for the mid-term examination in IFT2015, which is also the first part of the *examen pré-doctoral* in data structures.

- ◆ Topics for a «B/A-» level are denoted by ★; ★★ denote somewhat more advanced topics for «A+/A» level.
- ★ The margin notes refer to the following books :
  - S** Sedgewick, R. *Algorithms in Java*, Parts 1–4, 3<sup>e</sup> édition (2003)
  - SW** Sedgewick, R. et K. Wayne. *Algorithms*, 4<sup>e</sup> édition (2011)
- ★ The class notes and links to Wikipedia articles are available on the webpage <http://ift2015a16.wordpress.com/>.
- ★ No documentation is allowed at the examen.

# E1 Principles of algorithm analysis

## References

- ▷ Sedgewick chapter 2; Sedgewick & Wayne section §1.4
- ▷ Note on the foundations: `notes01-recursion.pdf`.
- ▷ Note on algorithm analysis: `notes04-analysis.pdf`.

## Topics

- ★ Basic principles : worst case, best case, average case. S§2.1,2.2,2.7
- ★ Growth of common functions : constants, logarithms, polynomials, exponentials. Factorial ( $n!$ ), Fibonacci numbers ( $F_n = F_{n-1} + F_{n-2}$ ), harmonic numbers ( $H_n = \sum_{i=1}^n 1/i$ ) S§2.3
- ★ Notion of amortized cost.
- ★★ Proving amortized cost. Credit/debit method.
- ★ Asymptotic notation : definitions of big-Oh  $O(f)$ , small-oh  $o(f)$ ,  $\Theta(f)$ , and  $\Omega(f)$ . Arithmetic expressions involving asymptotics, rules :  $O(f) + O(g)$ ,  $O(f) \cdot O(g)$ . Connections to lim S§2.4

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c > 0 \quad \Rightarrow \quad f(n) = O(g(n));$$

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0 \quad \Leftrightarrow \quad f(n) = o(g(n));$$

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1 \quad \Leftrightarrow \quad f(n) \sim g(n)$$

- ★ Using the definitions to prove  $f = O(g)$  or  $f = o(g)$ .
- ★ Informal determination of space and time complexity for simple (iterative) algorithms
- ★ Basic recurrences. S§2.5,2.6

$$f(n) = f(n-1) + O(1) \qquad f(n) = O(n);$$

$$f(n) = f(n-1) + O(n) \qquad f(n) = O(n^2);$$

$$f(n) = f(n/2) + O(1) \qquad f(n) = O(\log n);$$

$$f(n) = f(n/2) + O(n) \qquad f(n) = O(n);$$

- ★★ Proof by induction for asymptotic recurrences.
- ★ Experimental validation of running time

## E2 Elementary structures and abstract data types

### References

- ▷ Sedgewick chapters 3 et 4 ; Sedgewick & Wayne sections 1.1–1.3
- ▷ Note on lists: `notes02-linkedlist.pdf`.
- ▷ Note on tables: `notes03-tableaux.pdf`.

### Topics

- ★ Java building blocks. S§3.1 ; SW§1.1
- ★ Tables. S§3.2
- ★ Linked lists. Variations : circular, doubly-linked lists. Sentinels for the head and/or tail. Manipulation of elements, insertion and deletion. List traversal. S§3.3,3.4
- ★ Memory management for lists. S§3.5
- ★ Concept of an abstract data type, interface, implementation, client. S§4.1 ; SW§1.2
- ★ Abstract types for stacks, queues and generalized queues, S§4.2,4.7
- ★ Implementations of stack and queue by tables or linked lists. Running time for standard operations in different implementations. Overflow/underflow. S§4.4,4.5,4.7 ; SW§1.3