

7. IFT2015 Structures de données: Liste détaillée de sujets¹

¹ Detailed List of Subjects for the Midterm Examination — English translation starts on Page 6

F0 Introduction

LE BUT DE CE DOCUMENT est de définir les compétences et connaissances requises dans le cours IFT2015 à l'examen intra. L'examen constitue également la première partie de l'examen pré-doctorale en structures de données (sous sigle IFT6002).

- ♦ 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 Sedgwick, R. *Algorithmes en Java*, 3^e édition (2004)
 - SW Sedgwick, R. et K. Wayne. *Algorithms*, 4^e édition (2011)
 - CLR Cormen, T., E. L. Leiserson, R. L. Rivest, et C. Stein. *Algorithmique*, 3^e édition.
- ★ Les notes de cours, présentations, et des liens vers ressources en-ligne sont affichés sur le site <http://ift2015a17.wordpress.com/>.
- ★ Aucune documentation ne sera permise à l'examen intra.



F1 Principes d'analyse d'algorithmes

Références

- ▷ Sedgewick chapitre 2
- ▷ Sedgewick & Wayne §1.4²
- ▷ Cormen, Leiserson, Rivest & Stein chapitres 1–3
- ▷ Notes sur les fondations : [handout01-recursion.pdf](#).
- ▷ Notes sur l'analyse d'algorithmes : [handout05-analysis.pdf](#).

² <http://algs4.cs.princeton.edu/14analysis/>

Sujets

- ★ Principes de base : pire cas, meilleur cas, moyen cas.
- ★ Croissance de fonctions communes : constantes, logarithmiques, polynomiales, exponentielles. Factorielle($n!$), approximation de Stirling³ nombres Fibonacci⁴, nombres harmoniques⁵.
- ★ Notation asymptotique⁶ : définitions de grand $O(f)$, petit $o(f)$, $\Theta(f)$ et $\Omega(f)$.

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

$$\begin{aligned} \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = c > 0 & \Rightarrow f(n) = O(g(n)); \\ \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0 & \Leftrightarrow f(n) = o(g(n)); \\ \lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 1 & \Leftrightarrow f(n) \sim g(n) \end{aligned}$$

- ★ Application de la définition pour démontrer $f = O(g)$ ou $f = o(g)$.
- ★ Détermination du temps de calcul et d'usage de mémoire pour algorithmes (itératifs) simples, et pour algorithmes récursifs (comme expression récursive).
- ★ Récurrences simples.

S§2.1,2.2,2.7; CLR 1

S§2.3

$$^3 n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

$$^4 F_0 = 0; F_1 = 1; F_n = F_{n-1} + F_{n-2} \{n > 1\}$$

$$^5 H_n = \sum_{i=1}^n 1/i = \ln n + \gamma + o(1)$$

⁶ [W_{\(fr\)}:comparaison asymptotique](#)

S§2.4; CLR 3

CLR 2

S§2.5.2.6

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

- ★★ Preuve par induction pour récurrences asymptotiques.
- ★ Notion de temps amorti.
- ★★ Preuves de résultats sur le coût amorti d'opérations. Principe d'analyse crédit/débit⁷.
- ★ Validation expérimentale de temps de calcul

CLR§17.4

⁷ [W_{\(en\)}:accounting method](#)

F2 Structures élémentaires et types abstraits

Références

- ▷ Sedgewick chapitres 3 et 4
- ▷ Sedgewick & Wayne §1.1⁸, §1.2⁹, §1.3¹⁰
- ▷ Cormen, Leiserson, Rivest & Stein §10.1, §10.2
- ▷ Notes sur les types abstraits : [handout02-tad.pdf](#).
- ▷ Notes sur les tableaux : [handout03-tableaux.pdf](#).
- ▷ Notes sur les listes : [handout04-chaining.pdf](#).

⁸ <http://algs4.cs.princeton.edu/11model/>

⁹ <http://algs4.cs.princeton.edu/12oop/>

¹⁰ <http://algs4.cs.princeton.edu/13stacks/>

Sujets

- ★ Blocs de construction pour programmes Java. S§3.1;SW§1.1
- ★ Notions de 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
- ★ Listes chaînées¹¹. Variations : listes circulaires, doublement chaînées. ¹¹ W_(fr):liste chaînée
- Sentinelles¹² pour la tête et/ou la queue. Manipulation d'éléments sur la S§3.3,3.4;CLR§10.2
- liste, insertion et suppression. Parcours d'une liste. ¹² W_(en):sentinel
- ★ Tableaux¹³. ¹³ W_(fr):tableau
- ★ Implantations de pile et de queue par tableaux ou listes chaînées. Efficacité d'implantations différentes (temps de calcul pour les opérations S§3.2
- standards). Débordement. S§4.4,4.5,4.7;SW§1.3;CLR§10.1

F3 Arbres

Références

- ▷ Sedgewick §4.3, §5.4–5.7
- ▷ Notes sur les listes et les arbres : [handout04-chaining.pdf](#).

Sujets

- ★ Algorithmes récursifs. Diviser pour régner. S§5.1,5.2
- ★ Terminologie pour structures arborescentes : arbre k -aire, hauteur, niveau, profondeur. Implémentation d'un arbre. S§5.4
CLR§10.4
- ★ Propriétés d'arbres binaires (relations entre le nombre de nœuds internes et externes ou la hauteur). S§5.5
- ★ Parcours d'un arbre : préfixe/préordre, infixe/dans l'ordre, postfixe/postordre, ordre de niveau. S§5.6
- ★ Arbre syntaxique. Conversions d'expressions arithmétiques : notations infixe, postfixe et préfixe. S§4.3
- ★ Algorithmes récursifs sur les arbres : calcul de taille, hauteur ou profondeur de sous-arbres. S§5.7

F4 Algorithmes sur graphes

Références

- ▷ Sedgewick §3.7
- ▷ Sedgewick & Wayne §4.1¹⁴, §4.2¹⁵
- ▷ CLR chapitre 22
- ▷ Notes sur les graphes : [handout06-graphes.pdf](#).

¹⁴ <http://algs4.cs.princeton.edu/41graph/>

¹⁵ <http://algs4.cs.princeton.edu/42digraph/>

Sujets

- ★ Représentation d'un graphe : matrice d'adjacence et listes d'adjacence¹⁶. S§3.7;SW§4.1;CLR§22.1
- ★ Parcours d'un graphe par profondeur et par largeur. S§5.8;SW§4.1;CLR§22.2,§22.3
- ★ Applications de parcours : composantes connexes, bipartition, tri topologique, plus courts chemins (à partir d'une source). SW§4.2;CLR§22.4

¹⁶ [W_{\(en\)}adjacency list](#)



E0 Introduction

THIS DOCUMENT defines the skills and knowledge for the midterm examination in IFT2015, which is also the first part of the *examen pré-doctoral* in data structures (as IFT6002).

- ◆ 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, 3rd edition (2003)
 - SW** Sedgewick, R. and K. Wayne. *Algorithms*, 4th edition (2011)
 - CLR** Cormen, T., E. L. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, 3rd edition.
- ★ The class notes, presentations, and links to online resources are available on the webpage <http://ift2015a17.wordpress.com/>.
- ★ No documentation is allowed at the examen.



E1 Principles of algorithm analysis

References

- ▷ Sedgewick chapter 2
- ▷ Sedgewick & Wayne §1.4¹⁷
- ▷ Cormen, Leiserson, Rivest & Stein chapters 1–3
- ▷ Notes on the foundations: [handout01-recursion.pdf](#).
- ▷ Notes on algorithm analysis: [handout05-analysis.pdf](#).

¹⁷ <http://algs4.cs.princeton.edu/14analysis/>

Topics

- ★ Basic principles : worst case, best case, average case.
- ★ Growth of common functions : constants, logarithms, polynomials, exponentials. Factorial ($n!$), Stirling's formula¹⁸, Fibonacci numbers¹⁹, harmonic numbers²⁰.
- ★ Asymptotic notation²¹ : 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§2.1,2.2,2.7; CLR 1

S§2.3

¹⁸ $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$

¹⁹ $F_n = F_{n-1} + F_{n-2}$

²⁰ $H_n = \sum_{i=1}^n 1/i = \ln n + \gamma + o(1)$

²¹ $W_{(en)}$: **big-O notation**

S§2.4; CLR 3

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

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

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

- ★ Using the definitions to prove $f = O(g)$ or $f = o(g)$.
- ★ Determination of space and time complexity for simple (iterative) algorithms, and for recursive algorithms (as a recursive expression).
- ★ Basic recurrences.

CLR 2

S§2.5,2.6

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

- ★★ Proof by induction for asymptotic recurrences.
- ★ Notion of amortized cost.
- ★★ Proving amortized cost. Credit/debit method.
- ★ Experimental validation of running time

CLR §17.4

*E2 Elementary structures and abstract data types**References*

- ▷ Sedgewick chapters 3 et 4
- ▷ Sedgewick & Wayne §1.1²², §1.2²³, §1.3²⁴
- ▷ Cormen, Leiserson, Rivest & Stein §10.1, §10.2
- ▷ Notes on linked lists: [handout02-linkedlist.pdf](#).
- ▷ Notes on tables: [handout03-tableaux.pdf](#).

²² <http://algs4.cs.princeton.edu/11model/>²³ <http://algs4.cs.princeton.edu/12oop/>²⁴ <http://algs4.cs.princeton.edu/13stacks/>*Topics*

- ★ Java building blocks.
- ★ Concept of an abstract data type, interface, implementation, client.
- ★ Abstract types for stacks, queues and generalized queues,
- ★ Linked lists²⁵. Variations : circular, doubly-linked lists. Sentinels²⁶ for head and/or tail. Manipulation of elements, insertion and deletion. List traversal.
- ★ Arrays²⁷.
- ★ Implementations of stack and queue by tables or linked lists. Running time for standard operations in different implementations. Overflow/underflow.

S§3.1;SW§1.1

S§4.1;SW§1.2;

S§4.2,4.7

²⁵ W_(en):linked list

S§3.3,3.4;CLR§10.2

²⁶ W_(en):sentinel²⁷ W_(en):array

S§3.2

S§4.4,4.5,4.7;SW§1.3;CLR§10.1

E3 Trees

References

- ▷ Sedgewick §4.3, §5.4–5.7
- ▷
- ▷ Notes on lists and trees: [handout04-chaining.pdf](#).

Topics

- ★ Recursive algorithms. Divide-and-conquer. S§5.1.5.2
- ★ Terminology for tree structures : k -ary tree, height, level, depth. Tree implementations. S§5.4
CLR §10.4
- ★ Mathematical properties of binary trees (relationships between number of internal and external nodes, height) S§5.5
- ★ Tree traversal : preorder, inorder, postorder, level-order. S§5.6
- ★ Syntax tree. Conversion between arithmetic notations : infix, prefix and postfix. S§4.3
- ★ Recursions on trees : computing the size, height, or depth of subtrees. S§5.7

E4 Graph algorithms

References

- ▷ Sedgewick §3.7
 - ▷ Sedgewick & Wayne §4.1²⁸, §4.2²⁹
 - ▷ Cormen, Leiserson, Rivest & Stein chapter 22
 - ▷ Notes on graphs: [handout06-graphes.pdf](#).
- ²⁸ <http://algs4.cs.princeton.edu/41graph/>
²⁹ <http://algs4.cs.princeton.edu/42digraph/>

Topics

- ★ Graph representations by adjacency matrix and adjacency lists³⁰. S§3.7;SW§4.1;CLR§22.1
- ★ Depth-first and breadth-first search (DFS and BFS) in a graph ³⁰ [W\(en\):adjacency list](#) S§5.8;SW§4.1;CLR§22.2
- ★ Applications of graph traversal : connected components, bipartite graph, topological sort, (single-source) shortest paths S§4.2;CLR§22.4