IFT2015 Structures de données: Liste détaillée de sujets[1]

Miklós Csűrös

10 décembre 2013

[1] Detailed List of Subjects for the Final Examination — English translation starts on Page 7

F0 Introduction

LE BUT DE CE DOCUMENT est de définir les compétences et connaissances requises dans le cours IFT2015 à l'examen final. L'examen constitue également la deuxième 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*, 3e édition (2004)

  SW Sedgewick, R. et K. Wayne. *Algorithms*, 4e édition (2011)

- ✶ Les notes de cours et des liens vers des articles Wikipedia sont affichés sur le site http://www.iro.umontreal.ca/~csuros/IFT2015/A13/.
- ✷ Aucune documentation ne sera permise à l'examen final.

F1 Principes d'analyse d'algorithmes

Références

- ▷ Sedgewick chapitre 2 ; Sedgewick & Wayne section §1.4
- ▷ Notes sur les fondations : `notes01-recursion.pdf`.
- ▷ Notes sur l'analyse d'algorithmes : `notes04-analysis.pdf`.
- ▷ Notes sur les aspects pratiques : `notes05-experiments.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[2], nombres Fibonacci[3], nombres harmoniques[4], logarithme itéré. S§2.3
- ⋆⋆ Fonction d'Ackermann et son inverse.
- ⋆ Notion de temps amorti.
- ⋆ 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 : S§2.4

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

[3] $F_n = F_{n-1} + F_{n-2}$

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

$O(f)+O(g)$, $O(f)\cdot O(g)$. Relations avec la limite

$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=c>0 \qquad \Rightarrow \qquad f(n)=O\big(g(n)\big);$$
$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=0 \qquad \Leftrightarrow \qquad f(n)=o\big(g(n)\big);$$
$$\lim_{n\to\infty}\frac{f(n)}{g(n)}=1 \qquad \Leftrightarrow \qquad f(n)\sim g(n)$$

- ⋆ Application 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

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

- ⋆ 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
- ▷ Notes sur les listes : `notes02-linkedlist.pdf`.
- ▷ Notes 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

F3 Arbres

Références

- ▷ Sedgewick sections 4.3, 5.4–5.7
- ▷ Notes sur les arbres : `notes06-trees.pdf`.

Sujets

- ⋆ Terminologie pour structures arborescentes : arbre k-aire, hauteur, niveau, profondeur. Implémentation d'un arbre. S§5.4
- ⋆ 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 Appartenance-union

Références

- ▷ Sedgewick sections 1.2–1.3 ; Sedgewick & Wayne section 1.5
- ▷ Notes sur Union-Find : `notes07-unionfind.pdf`.

Sujets

- ⋆ Problème de connexité, opérations d'appartenance-union. S§1.2
- ⋆ Structure Union-Find. Astuces : union-par-rang/union-par-taille, compression de chemin. S§1.3 ;SW§1.5
- ⋆⋆ Coût amorti d'opérations : $O(\alpha(m,n))$ pour Union-Find avec union équilibrée et compression de chemin

F5 File de priorité

Références

- ▷ Sedgewick sections 9.1–9.6 ; Sedgewick & Wayne section 2.4
- ▷ Notes sur les files à priorités : `notes08-heap.pdf`.
- ▷ Notes sur le tri par tas : `notes08b-heapsort.pdf`.

Sujets

- ⋆ Type abstrait de file de priorité min-tas/max-tas : opérations insert, deleteMin ou deleteMax. Implantations par tableau ou liste chaînée. S§9.1,9.5
- ⋆ Arbre en ordre de tas. Manipulation du tas : nager et couler (heapisation montante et descendante). Tas binaire, sa représentation dans un tableau. S§9.2,9.3 ;SW§2.4
- ⋆ heapify (établissement de l'ordre de tas dans un tableau) ; tri par tas, son temps de calcul et usage de mémoire. S§9.4

F6 Algorithmes sur graphes

Références

- ▷ Sedgewick sections 3.7, 5.8 ; Sedgewick & Wayne sections 4.1, 4.3
- ▷ Notes sur l'arbre couvrant minimal et le plus court chemin : `notes09-acm.pdf`.
- ▷ Notes sur le parcours de graphes : `notes10-dfs.pdf`.

Sujets

- ⋆ Représentation d'un graphe : matrice d'adjacence et listes d'adjacence. S§3.7 ;SW§4.1
- ⋆ Parcours d'un graphe par profondeur et par largeur. S§5.8 ;SW§4.1
- ⋆ Notion d'un arbre couvrant minimal. Principe de base des algorithmes : la règle bleue. Logique générale des algorithmes de Prim et de Kruskal, choix de structures de données. SW§4.3
- ⋆⋆ Analyse détaillé du temps de calcul des algorithmes. Avantages d'un tas d-aire ou Fibonacci dans l'algorithme de Prim.

F7 Méthodes de tri

Références

- ▷ Sedgewick sections 6.1–6.4, 6.6, 6.9 ; chapitres 7, 8.
- ▷ Sedgewick & Wayne sections 2.1, 2.2, 2.3.
- ▷ Notes sur les tris : `notes12-tris.pdf`.
- ▷ Notes sur le tri rapide : `notes13-quicksort.pdf` `notes14-quicksort2.pdf`
- ▷ Notes sur les permutations : `notes14b-permutations.pdf`.

Sujets

- ⋆ Terminologie : tri stable, tri interne et externe. S§6.1
- ⋆ Tri par sélection et tri par insertion. S§6.3,6.4 ; SW§2.1
- ⋆ Performances des tris élémentaires (pire cas, meilleur cas, cas moyen). S§6.6
- ⋆ Fusion de tableaux. S§8.1
- ⋆ Tri par fusion (descendant), sa performance. S§8.3,8.4 ; SW§2.2
- ⋆ Tri rapide : algorithme de base. Améliorations : partition par la médiane-de-trois, petits sous-fichiers. S§7.1,7.4,7.5 ; SW§2.3
- ⋆ Génération d'une permutation aléatoire
- ⋆ Performances du tri rapide (pire cas, meilleur cas, cas moyen) S§7.2,7.3 ; SW 2.3
- ⋆⋆ Preuve de la performance moyenne $O(n \log n)$ du tri rapide.
- ⋆ Preuve de la borne inférieure $\lg(n!)$ sur le nombre de comparaisons au pire pour trier SW§2.2

F8 Arbres binaires de recherche

Références

- ▷ Sedgewick chapitres 12, sections 13.1, 13.3, 13.4
- ▷ Sedgewick & Wayne sections 3.1, 3.2.
- ▷ Notes sur les arbres binaires de recherche : `notes15-abr.pdf`.

▷ Notes sur les arbres rouge-et-noir : `notes16-rn.pdf`.
▷ Notes sur les arbres 2-3-4 : `notes17-234.pdf`.

Sujets

- ⋆ Type abstrait de la table de symboles. S§12.1,12.2
- ⋆ Recherche séquentielle et recherche binaire. S§12.3–12.5
- ⋆ Arbre binaire de recherche. Procédures fondamentales sur un ABR : recherche, insertion, suppression. Recherche de minimum ou maximum, successeur ou prédecesseur. S§12.6–12.9
- ⋆ Performance moyenne des opérations sur un ABR standard avec clés aléatoires. S§13.1
- ⋆ Notion d'un ABR équilibré. Maintenance d'équilibre : rotations simples et doubles.
- ⋆ ABR rouge et noir. Définition par rang (hauteur noire) ou coloriage ; équivalence des deux définitions. Coût des opérations dans le pire cas. S§13.4
- ⋆⋆ Hauteur maximale d'un arbre rouge et noir.
- ⋆ Techniques de base sur les ABR rouges et noirs : promotion/rétrogradation, changement de couleur, rotation. Déroulement général d'une insertion ou suppression.
- ⋆⋆ Déroulement détaillé de l'insertion et de la suppression.
- ⋆ Les arbres 2-3-4, et leur équivalence avec les arbres rouges et noirs. Techniques de base sur les arbres 2-3-4 : décalage et découpage, leur relation aux rotations et promotions. S§13.3

F9 Tableaux de hachage

Références

▷ Sedgewick chapitre 14.
▷ Sedgewick & Wayne sections 3.4, 3.5.
▷ Notes sur le hachage : `notes18-hashing.pdf`.

Sujets

- ⋆ Notions de base pour tableaux de hachage : facteur de charge/remplissage, collisions. S§14.1
- ⋆ Fonctions de hachage : méthodes de la division et de la multiplication.
- ⋆⋆ Hachage universel.
- ⋆ Résolution de collisions par chaînage séparé. Coût moyen des opérations de l'interface (table de symboles) en fonction de la facteur de charge. S§14.2
- ⋆ Addressage ouvert : notion de sondage/test. Procédures de recherche et d'insertion avec addressage ouvert. Suppression paresseuse et hachage dynamique. Sondage linéaire, grappe forte. Double hachage. S§14.3–14.6
- ⋆⋆ Coût moyen des opérations de l'interface avec sondage linéaire et double hachage en fonction de la facteur de charge.

E0 Introduction

THIS DOCUMENT defines the skills and knowledge for the final examination in IFT2015, which is also the second 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, 3rd edition (2003)

  SW Sedgewick, R. et K. Wayne. *Algorithms*, 4th edition (2011)

- ✶ The class notes and links to Wikipedia articles are available on the webpage http://www.iro.umontreal.ca/~csuros/IFT2015/A13/.
- ✷ No documentation is allowed at the examen.

E1 Principles of algorithm analysis

References

- ▷ Sedgewick chapter 2 ; Sedgewick & Wayne section §1.4
- ▷ Notes on the foundations: `notes01-recursion.pdf`.
- ▷ Notes on algorithm analysis: `notes04-analysis.pdf`.
- ▷ Notes on practical aspects: `notes05-experiments.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 ($n!$), Stirling's formula[5], Fibonacci numbers[6], harmonic numbers[7], iterated logarithm S§2.3
- ⋆ Notion of amortized cost.
- ⋆⋆ Ackermann's function and its inverse
- ⋆ 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

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = c > 0 \qquad \Rightarrow \qquad f(n) = O\big(g(n)\big);$$

$$\lim_{n\to\infty}\frac{f(n)}{g(n)} = 0 \qquad \Leftrightarrow \qquad f(n) = o\big(g(n)\big);$$

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

- ⋆ Using the definitions to prove $f = O(g)$ or $f = o(g)$.
- ⋆ Informal determination of space and time complexity for simple (iterative) algorithms

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

[6] $F_n = F_{n-1} + F_{n-2}$

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

- ⋆ Basic recurrences. S§2.5,2.6

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

- ⋆⋆ 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
- ▷ Notes on lists: `notes02-linkedlist.pdf`.
- ▷ Notes 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

E3 Trees

References

- ▷ Sedgewick sections 4.3, 5.4–5.7
- ▷ Notes on trees: `notes06-trees.pdf`.

Topics

- ⋆ Terminology for tree structures : k-ary tree, height, level, depth. Tree implementations. S§5.4
- ⋆ 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 Union-find

References

- ▷ Sedgewick sections 1.2–1.3 ; Sedgewick & Wayne section 1.5
- ▷ Notes on Union-Find: `notes07-unionfind.pdf`.

Topics

- ⋆ Connectivity problems, union-find operations. S§1.2
- ⋆ Union-Find data structure. Techniques : union-by-rank/union-by-size, path compression. S§1.3 ;SW§1.5
- ⋆⋆ Amortized cost per operation : $O(\alpha(m,n))$ for Union-Find with balanced trees and path compression.

E5 Priority queues

References

- ▷ Sedgewick sections 9.1–9.6 ; Sedgewick & Wayne section 2.4
- ▷ Notes on priority queues: `notes08-heap.pdf`.
- ▷ Notes on heapsort: `notes08b-heapsort.pdf`.

Topics

- ⋆ ADT for priority queue : operations insert, deleteMin or deleteMax. Implementations by table or linked list. S§9.1,9.5
- ⋆ Heap order for a tree. Heap manipulation : swim and sink. Binary heap, its representation in a table. S§9.2,9.3 ;SW§2.4
- ⋆ heapify (linear-time construction of heap order in a table) ; Heapsort, its running time and memory. S§9.4

E6 Graph algorithms

References

- ▷ Sedgewick sections 3.7, 5.8 ; Sedgewick & Wayne sections 4.1, 4.3
- ▷ Notes on minimum spanning tree and shortest path: `notes09-acm.pdf`.
- ▷ Notes on graph traversal: `notes10-dfs.pdf`.

Topics

- ⋆ Graph representations by adjacency matrix and adjacency lists. S§3.7 ;SW§4.1
- ⋆ Depth-first and breadth-first search in a graph S§5.8 ;SW§4.1
- ⋆ Concept of a minimal spanning tree (MST). Basic principles of MST algorithms : the blue rule (adding minimum-weight edge in a cut). General logic of Kruskal's and Prim's algortithms, choice of data structures. Basic justification for $O(n \log n)$ and $O(m \log n)$ running times (n nodes, m edges). SW§4.3
- ⋆⋆ Detailed analysis of running time for Kruskal's and Prim's algorithms. Uses of d-ary or Fibonacci heap in Prim's algorithm.

E7 Sorting algorithms

References

- ▷ Sedgewick Sections 6.1–6.4, 6.6, 6.9 ; Chapters 7, 8.
- ▷ Sedgewick & Wayne sections 2.1, 2.2, 2.3.
- ▷ Notes on elementary sorting algorithms: `notes12-tris.pdf`.
- ▷ Notes on quicksort : `notes13-quicksort.pdf` `notes14-quicksort2.pdf`
- ▷ Notes sur permutations : `notes14b-permutations.pdf`.

Topics

⋆ Terminology : stable sort, internal and external sort.	S§6.1
⋆ Insertion sort and selection sort.	S§6.3,6.4 ; SW §2.1
⋆ Performance of elementary sorting algorithms (worst case, best case, average case).	S§6.6
⋆ Merging arrays.	S§8.1
⋆ Mergesort (top-down), its performance.	S§8.3,8.4 ; SW§2.2
⋆ Quicksort : basic algorithm. Improvements : pivoting by median-of-three, small subarrays.	S§7.1,7.4,7.5 ; SW§2.3
⋆ Performance of quicksort (worst case, best case, average case).	S§7.2,7.3
⋆ Generating a random permutation	
⋆⋆ Proof of $O(n \log n)$ average running time for quicksort.	
⋆ Proof of the lower bound $\lg(n!)$ for the worst-case number of comparisons	SW§2.2

E8 Binary search trees

Reference

- ▷ Sedgewick chapters 12, sections 13.1, 13.3 and 13.4
- ▷ Sedgewick & Wayne sections 3.1, 3.2.
- ▷ Notes on binary search trees: `notes15-abr.pdf`.
- ▷ Notes on red-black trees: `notes16-rn.pdf`.
- ▷ Notes on 2-3-4 trees: `notes17-234.pdf`.

Topics

⋆ Abstract data type of symbol table.	S§12.1,12.2
⋆ Sequential and binary search.	S§12.3–12.5
⋆ Binary search tree. Basic techniques : search, insertion, deletion. Searching for minimum or maximum, successor or predecessor.	S§12.6–12.9
⋆ Average performance of a standard BST with random keys.	S§13.1
⋆ Notion of a balanced BST. Maintaining the balance : simple and double rotations.	
⋆ Red-black tree. Definition by rank (black height) or coloring ; equivalence of the two definitions. Time complexity for operations in the worst-case.	S§13.4

- ⋆⋆ Maximum height of a red-black tree.
- ⋆ Basic techniques for red-black trees : promotion/demotion, recoloring, rotations. General outline of insertion and deletion.
- ⋆⋆ Detailed (case-by-case) steps in insertion and deletion.
- ⋆ 2-3-4 trees, their equivalence with red-black trees. Basic techniques with 2-3-4 trees : shifting and splitting, relationship with promotions and rotations in red-black tree. S§13.3

E9 Hash tables

Reference

- ▷ Sedgewick chapter 14.
- ▷ Sedgewick & Wayne sections 3.4, 3.5.
- ▷ Notes on hashing: `notes18-hashing.pdf`.

Topics

- ⋆ Basic notions for hashtables : load factor, collisions. S§14.1
- ⋆ Hash functions : division and multiplication methods.
- ⋆⋆ Universal hashing.
- ⋆ Collision resolution by separate chaining. Average-case performance with separate chaining as function of the load factor. S§14.2
- ⋆ Open addressing : probe sequence. Search and insertion with open addressing. Lazy deletion, dynamic hashing. Linear probing, primary clustering. Double hashing. S§14.3–14.6
- ⋆⋆ Average-case performance of linear probing and double hashing as function of the load factor.