IFT2015 Structures de données
Liste détaillée de sujets

Miklós Csűrös

Département d'informatique et de recherche opérationnelle
Université de Montréal

Hiver 2011

F0 Introduction

Le but de ce document est de définir les connaissances requises dans le cours IFT2015 à l'examen final qui constitue aussi la deuxième partie de l'examen prédoctorale 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 au livre de Sedgewick [*Algorithmes en Java*]. Les fichiers pour les notes de cours se trouvent à l'URL `http://www.iro.umontreal.ca/~csuros/IFT2015/H11/materiel/`.

F1 Principes d'analyse d'algorithmes

Références

- ▷ Sedgewick chapitre 2
- ▷ Notes sur la notation asymptotique : `notes03-croissance.pdf`.
- ▷ Notes sur les fondations mathématiques : `notes03A-bigO.pdf`.
- ▷ Notes sur l'analyse d'algorithmes : `notes04-analyse.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
- ⋆ Notation asymptotique : définitions de grand $O(f)$, petit $o(f)$, $\Theta(f)$ et $\Omega(f)$. Expressions avec $O()$ ou $o()$, règles d'arithmétique : $O(f) + O(g)$, $O(f) \cdot O(g)$. Relations avec la limite S§2.4

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

  Comparaison de fonctions de forme $f(n) = \sum_i a_i b_i^n n^{c_i} (\lg n)^{d_i}$.
- ⋆ Techniques de preuve. Application directe de la définition pour démontrer $f = O(g)$ ou $f = o(g)$.
- ⋆ Récurrences de base. S§2.5,2.6

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

- ⋆ Variations des récurrences de base : «deviner» l'ordre de grandeur [méthode de substitutions] en termes de notation $O()$; preuve par induction.
- ⋆⋆ Solution de récurrences par substitution de variables.
- ⋆ Notion de coût amorti.

F2 Structures élémentaires

Références

- ▷ Sedgewick chapitre 3
- ▷ Notes sur les listes : `notes01-linkedlist.pdf`.
- ▷ Notes sur les tableaux : `notes02-tableaux.pdf`.

Sujets

- ⋆ Blocs de construction pour programmes Java. S§3.1
- ⋆ 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
- ⋆ Structures endogènes et exogènes. Gestion de mémoire pour listes. S§3.5

F3 Types abstraits

Références

- ▷ Sedgewick chapitre 4.
- ▷ Notes sur les listes : `notes01-linkedlist.pdf`.
- ▷ Notes sur les tableaux : `notes02-tableaux.pdf`.

Sujets

- ⋆ Notion d'un type abstrait, interface, implantation, client. S§4.1
- ⋆ Types abstraits de files généralisées, piles et queues/files FIFO. S§4.2,4.7
- ⋆ Implémentations 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

F4 Récursion et arbres

Références

- ▷ Sedgewick chapitre 5.
- ▷ Notes sur l'analyse d'algorithmes : `notes04-analyse.pdf`.
- ▷ Notes sur les récurrences : `notes05-recursion.pdf`.
- ▷ Notes sur les arbres : `notes06-arbres.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
- ⋆ 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

F5 File de priorité

Références

- ▷ Sedgewick sections 9.1–9.5.
- ▷ Notes sur les files à priorités : `notes07-heap.pdf`.
- ▷ Notes sur le tri par tas : `notes08-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 sombrer (heapisation montante et descendante). Tas binaire, sa représentation dans un tableau. S§9.2,9.3
- ⋆⋆ Tas d-aire.
- ⋆ heapify (établissement de l'ordre de tas dans un tableau). S§9.4

F6 Méthodes de tri

Référence

- ▷ Sedgewick sections 6.1–6.4, 6.6, 3.4, 6.9, 10.2 ; chapitres 7, 8.
- ▷ Notes sur les tris élémentaires : `notes09-tris.pdf`.
- ▷ Notes sur le tri par fusion : `notes11-fusion.pdf`.
- ▷ Notes sur le tri rapide : `notes12-quicksort.pdf`.
- ▷ Notes sur le tri par tas : `notes08-heapsort.pdf`.

Sujets

- ⋆ Terminologie : tri stable, tri interne et externe. S§6.1
- ⋆ Tri par sélection et tri par insertion. S§6.3,6.4
- ⋆ Performances des tris élémentaires (pire cas, meilleur cas, cas moyen). S§6.6
- ⋆ Tri de liste chaînée. S§3.4,6.9
- ⋆ Tri rapide : algorithme de base. Améliorations : partition par la médiane-de-trois, petits sous-fichiers. S§7.1,7.4,7.5
- ⋆ Performances du tri rapide (pire cas, meilleur cas, cas moyen) S§7.2,7.3
- ⋆⋆ Preuve de la performance moyenne $O(n \log n)$ du tri rapide.
- ⋆ Fusion de tableaux. S§8.1
- ⋆ Tri par fusion (descendant), sa performance. S§8.3,8.4
- ⋆ Tri par tas, son temps de calcul et usage de mémoire. §9.4
- ⋆ Tri par le bit le plus significatif. §10.2

F7 Arbres binaires de recherche

Référence

- ▷ Sedgewick chapitres 12 et 13, sauf 13.5 (listes à sauts).
- ▷ Notes sur les arbres binaires de recherche : `notes13-abr.pdf`.
- ▷ Notes sur splaying : `notes14-splay.pdf`.
- ▷ Notes sur les arbres rouge-et-noir : `notes15-rn.pdf`.
- ▷ Notes sur les arbres 2-3-4 : `notes16-234.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 (cas simple et le cas d'un nœud à deux enfants non-null). 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. Définition d'équilibre dans un arbre AVL.
- ⋆⋆ Hauteur minimale d'un arbre AVL.
- ⋆ ABR *splay*. Principle de déploiement (*splaying*). Coût amorti des opérations de l'interface (table de symboles). S§13.2
- ⋆⋆ Règles de déploiement.
- ⋆ ABR rouge et noir. Définition par rang ou coloriage ; équivalence des deux définitions. Coût des opérations de l'interface dans le pire cas. S§13.4
- ⋆⋆ Hauteur minimale 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

F8 Tableaux de hachage

Référence

- ▷ Sedgewick chapitre 14.
- ▷ Notes de cours sur le hachage : `notes17-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, phénomène de 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

Topics for a «B/A-» level are denoted by $\star$; $\star\star$ denote somewhat more advanced topics for «A+/A» level. The margin notes refer to Sedgewick's *Algorithms in Java*. The class notes are available on the webpage `http://www.iro.umontreal.ca/~csuros/IFT2015/H11/materiel/`.

E1 Principles of algorithm analysis

References

- ▷ Sedgewick chapter 2
- ▷ Notes on asymptotic notation : `notes03-croissance.pdf`.
- ▷ Notes on the mathematical foundations : `notes03A-bigO.pdf`.
- ▷ Notes on algorithm analysis : `notes04-analyse.pdf`.

Topics

- $\star$ Basic principles : worst case, best case, average case. S§2.1,2.2,2.7
- $\star$ Growth of common functions : constants, logarithmes, polynomes, 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
- $\star$ 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

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

  Comparing functions in the form $f(n) = \sum_i a_i b_i^n n^{c_i} (\lg n)^{d_i}$.
- $\star$ Proof techniques. Using the definitions to prove $f = O(g)$ or $f = o(g)$.
- $\star$ Basic recurrences. S§2.5,2.6

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

- ⋆ Variations on basic recurrences : "guessing" the asymptotics [iterated substitutions], proof by induction.
- ⋆⋆ Variable substitution for solving recursions.
- ⋆ Notion of amortized cost.

E2 Elementary structures

References

- ▷ Sedgewick chapter 3
- ▷ Notes on lists : `notes01-linkedlist.pdf`.
- ▷ Notes on tables : `notes02-tableaux.pdf`.

Topics

- ⋆ Java building blocks. S§3.1
- ⋆ 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
- ⋆ Endogeneous and exogeneous structures. Memory management for lists. S§3.5

E3 Abstract data types

References

- ▷ Sedgewick chapter 4.
- ▷ Notes on lists : `notes01-linkedlist.pdf`.
- ▷ Notes on tables : `notes02-tableaux.pdf`.

Topics

- ⋆ Concept of an abstract type, interface, implementation, client. S§4.1
- ⋆ 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

E4 Recursion and trees

Reference

- ▷ Sedgewick chapter 5.
- ▷ Notes on algorithm analysis : `notes04-analyse.pdf`.
- ▷ Notes on recursion : `notes05-recursion.pdf`.
- ▷ Notes on trees : `notes06-arbres.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
- ⋆ 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

E5 Priority queues

References

- ▷ Sedgewick sections 9.1–9.5.
- ▷ Notes on heaps : `notes07-heap.pdf`.

▷ Notes on heapsort : `notes08-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. Implementation of decreaseKey. S§9.2,9.3
- ⋆⋆ d-ary heap.
- ⋆ heapify (linear-time construction of heap order in a table). S§9.4

E6 Sorting algorithms

References

- ▷ Sedgewick Sections 6.1–6.4, 6.6, 3.4, 6.9, 10.2 ; Chapters 7, 8.
- ▷ Notes on elementary sorting algorithms : `notes09-tris.pdf`.
- ▷ Notes on mergesort : `notes11-fusion.pdf`.
- ▷ Notes on quicksort : `notes12-quicksort.pdf`.
- ▷ Notes on heapsort : `notes08-heapsort.pdf`.

Topics

- ⋆ Terminology : stable sort, internal and external sort. S§6.1
- ⋆ Insertion sort and selection sort. S§6.3,6.4
- ⋆ Performance of elementary sorting algorithms (worst case, best case, average case). S§6.6
- ⋆ Sorting a linked list. S§3.4,6.9
- ⋆ Quicksort : basic algorithm. Improvements : pivoting by median-of-three, small subarrays. S§7.1,7.4,7.5
- ⋆ Performance of quicksort (worst case, best case, average case). S§7.2,7.3
- ⋆⋆ Proof of $O(n \log n)$ average running time for quicksort.
- ⋆ Merging arrays. S§8.1
- ⋆ Mergesort (top-down), its performance. S§8.3,8.4
- ⋆ Heapsort, its running time and memory usage. §9.4
- ⋆ Radix sort by most significant digit. §10.2

E7 Binary search trees

Reference

- ▷ Sedgewick chapters 12 and 13, except 13.5 (skip lists).
- ▷ Notes on binary search trees : `notes13-abr.pdf`.
- ▷ Notes on splaying : `notes14-splay.pdf`.
- ▷ Notes on red-black trees : `notes15-rn.pdf`.
- ▷ Notes 2-3-4 trees : `notes16-234.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.	
⋆ Notion of a balanced BST. Maintaining the balance : simple and double rotations.	S§13.1
⋆ Splay trees. Principle of splaying. Amortized cost of operations.	S§13.2
⋆⋆ Splaying rules.	
⋆ Red-black tree. Definition by rank 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

E8 Hash tables

Reference

- ▷ Sedgewick chapter 14.
- ▷ Notes on hashing : `notes17-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.