The digital tree process

Julien Clément, Mark Daniel Ward

The analysis of trees was integrated in Philippe Flajolet's writings and research throughout his life. In particular, he had a very keen interest in the *digital tree process* [PF034; PF187], omnipresent in Computer Science, for which the trie data structure is the strongest and best known embodiment. During Philippe's invited lecture at STACS'06, which was devoted to tries, he said that the treatment "cannot be but a brief guide to a rich subject whose proper development would require a book of full length" [PF187].

1. A central role in computer science

The digital tree process is found practically anywhere that data is classified or sorted and has abundant applications.

In a nutshell, this process relies on the principle of the thumb rule in dictionaries. As a data structure, the most pervasive kind of digital tree is the re*trie*val tree, introduced by de la Briandais [Bri59] and Fredkin [Fre60] [1], and usually shortened to "trie". (Section 6.3 of Knuth [Knu98] is a very helpful and fundamental discussion of the fundamentals, which traces the principles of tries to Thue [Thu12].) A partitioning of the data items—often using a sorting or classification by types—takes place at the root node. The tree is built recursively, according to subsequent bits or digits of the data. The children of the root are sorted further into subtrees and are thus partitioned more finely. The data items (also known as keys or strings) eventually require no more sorting and are ultimately stored in leaves of the trie. Due to their generality, tries are one of the most widely-known and greatly-studied data structures for representing a

1. As a side note, let us mention that initially Fredkin intended that tries should be pronounced "tree" as in the word "re*trie*val". Alas the trie data structure escaped its creator and, nowadays, people mostly pronounced it as the word "try", to distinguish verbally from "tree".

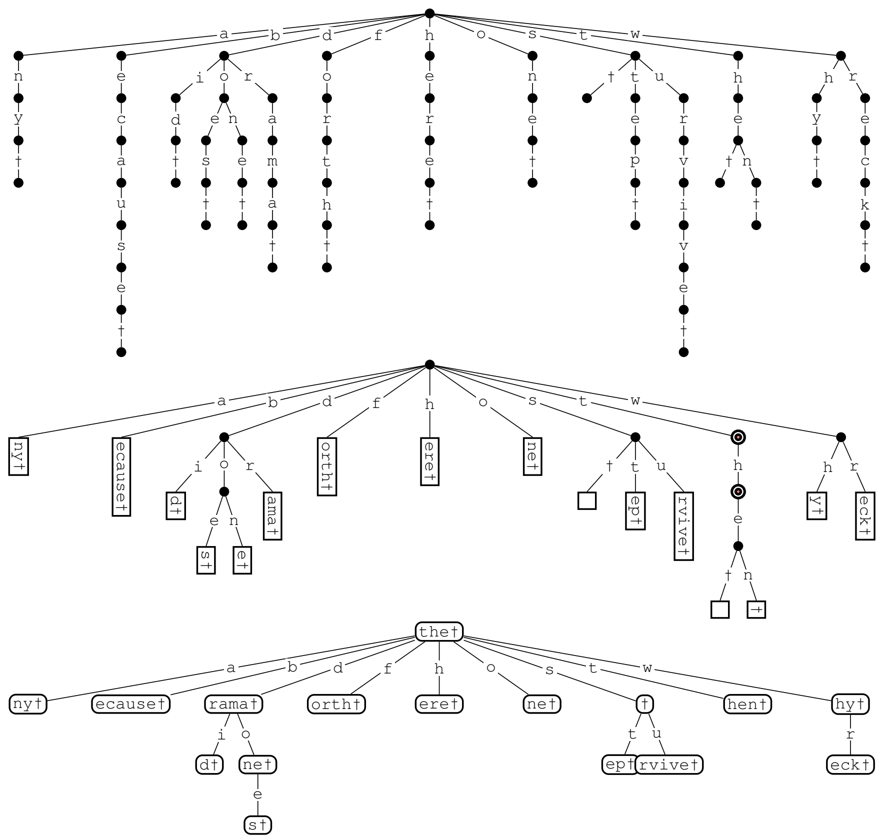

Figure 0.1. From top to bottom, a fully developed digital tree, a trie and a digital search tree built upon the words from the last sentence of the novel *Moby Dick* by H. Melville (inserted in order – relevant only for the DST): *"The Drama's Done. Why then here does any one step forth? - Because one did survive the wreck."*. In the trie, unary nodes which would not be present in PATRICIA tries are double circled. A terminal symbol '†' is added to each word inserted.

set of words. The trie data structure allows for all basic algorithmic operations one can expect from a dynamic dictionary-type data structure (inserting, deleting, searching, enumerating) and can also be used to sort a set of strings.

More formally, given an alphabet $\mathcal{A} = \{a_1, \dots, a_r\}$ of cardinality r, and for a prefix-free[2] set of words Y with letters from $\mathcal{A}$, the trie $T(Y)$ associated to Y is

2. The prefix-free property just means that none of the words is prefix of another.

defined recursively thanks to the following rules

$$T(Y) := \begin{cases} \emptyset, & \text{if } Y = \emptyset; \\ \sigma, & \text{if } Y = \{\sigma\}; \\ \langle \bullet, T(Y \setminus a_1), T(Y \setminus a_2), \cdots T(Y \setminus a_t) \rangle, & \text{otherwise.} \end{cases}$$

where $\bullet$ denotes an internal node and $Y \setminus \alpha$ is the subset built from Y by considering words that start with the letter α, and stripped of their initial symbol α. For a trie, the recursion stops as soon as Y contains less than two elements. Thus in order to build the trie for Y, one needs to consider only the minimal set of prefixes from Y by which all words are distinguished one from another. The prefix-freeness condition is merely technical and is easily forced by adding a terminal symbol (not belonging to the alphabet) to each string. (If two strings have the same finite length, these terminal symbols might need to be different to avoid collisions of two identical strings; e.g., see the two occurrences of the word `one` in the Moby Dick example.) The abstract data structure has given rise to many algorithmic variants. For instance PATRICIA tries [3] [Mor68] are tries where only "useful internal nodes", i.e., participating in the branching process, are considered. Thus in this setting unary nodes (with only one child) are removed. At the very end of the spectrum lies the digital search tree (DST): it uses the same partitioning process as tries but strings are stored inside internal nodes (like Binary Search Trees). Unlike the case of tries, a DST for a set of words Y depends on the order in which the strings are inserted. So loosely speaking, digital search trees are intermediary between tries and binary search trees. When coming to implementation of digital trees, even for "usual" tries, several options are possible depending on the decision structure chosen to guide descent in each node to subtrees (arrays of pointers, linked lists, or binary search trees for instance). Some variants of digital trees have been precisely analyzed by Philippe Flajolet [PF061; PF140; PF161].

With a more conceptual point of view, the structure of a trie can be used to model or analyze the behavior of both deterministic and stochastic algorithms in computer science. Tries are especially relevant to branching and sorting processes. So it is not surprising that the digital tree process has ramifications in the management of large databases (dynamic hashing [PF050], see also the introduction to FIXTHIS-the Chapter on Hashing; probabilistic counting [PF037], see FIXTHIS-Chapter on Approximate Counting), in communication protocols [PF065] (for instance, for leader election [KMW11; Pro93]), data compression (Lempel-Ziv and its variants [ZL77], suffix trees [Fay04; LSW07]), pattern matching [BCN12; JS94], random generation (to analyse precise schemes [PF042] or used as auxiliaries in Buffon machines [PF200]) and finally, rather unexpectedly, in computational geometry (for exact comparison of rationals [PF157]).

The digital tree process is elegant and simple. In its algorithmic form, it is intuitive to implement and utilize. This helps explain why the algorithmic, analytic and probabilistic aspects of digital trees are fundamental in both theoretical and applied domains in computer science.

3. For "Practical Algorithm to Retrieve Information Coded in Alphanumeric".

2. Digital trees in Philippe Flajolet's works

We can identify three main periods with respect to the study of digital trees by Philippe Flajolet.

The first period corresponds roughly to the early 1980s and focuses on tries and their applications in computer science. One must remember that, at this time of precursors, the average case analysis of algorithms was not yet a well recognized field among computer scientists (despite the spreading of Knuth's ideas). Worst-case analysis—focusing on pathological cases—were then the norm to study efficiency of algorithms. Tries (especially binary tries) and the underlying dyadic partition process were (and still are) ideal tools for demonstrating the utility of average case analysis. Indeed, although the trie data structure is very efficient on average (and, indeed, can compete with the best known data structures in many applications), the worst case complexity is unbounded[4]! It is worth noting that these first papers [PF034; PF039; PF053] introduce, in a pedagogical way, the case of binary tries built on a set of finite words of the same length ℓ (choosing n words amongst the 2^ℓ possible ones). This is a first combinatorial model easy to grasp. Of course it is not surprising that this model coincides—when ℓ tends to infinity—with the trie model for infinite binary strings, which has proved to be the natural framework for the analysis of tries; this framework is also rigorous, using a probability measure discussed in [Pit85]. In view of the applications (for instance, dynamic hashing or sorting), a general and symbolic methodology for average case analysis of digital trees is presented, with a distinction made between additive parameters such as size or path length, and extremal (or multiplicative) parameters like the height.

The second period, during the mid 1980's to mid 1990's, Philippe Flajolet's work on digital trees is more oriented towards the refinement of methodological tools for studying variations of tries (that are numerous: multi-way trie, PATRICIA trie, quadtrie, k dimensional trie, LC-trie, ternary search tries, Digital search tree, etc). What was certainly appealing to Philippe Flajolet is that these analyses lead to challenging mathematical problems. He sharpened and generalized several analytic tools by working with digital trees. Amongst these many techniques used by Philippe and his co-authors to analyze digital trees and their variants [PF039; PF053; PF101; PF120], we emphasize his pioneering work to systematically apply generating functions, the symbolic method, singularity analysis, the saddle point method, the Mellin transform, and poissonization/depoissonization techniques. Using tries and variants, he made many fruitful incursions into domains such as polynomial factorization [PF036], algebraic methods [PF053], differential equations [PF101], and random number generation [PF042].

For roughly the last 12 years of his life (1999–2011), he paid particular attention to developing and analyzing general probabilistic frameworks and tools [PF161; PF187; PF202; PF208], characterizing the stochastic generation of words and strings

4. Intuitively two words in a trie can share a prefix of arbitrary length; fortunately, the probability of having any infinite-length overlaps is 0 in the naturally induced probability measure.

which are inserted in digital trees [5]. This aspect is clearly related to information theory (see Chapter FIXTHIS-XYZ on Information Theory). Indeed one fundamental aspect of random digital trees is that the randomness essentially comes from the set of words they are built on; hence, there is a need to precisely model the source producing (infinite) words. This need was resolved by a very general framework introducing dynamical sources (mainly designed by Brigitte Vallée; see the corresponding chapter/volume-FIXTHIS-on-Dynamic-Sources for related works of Philippe Flajolet), which relies on transfer operators of dynamical systems theory. In this framework various parameters of tries are analysed [PF161] showing that the constants involved are related to intrinsic properties of the source like the entropy of the source. The *Grail* of such a study would be to consider for analyses a totally general framework where a (stochastic) source is induced from the family of fundamental probabilities $\{p_w\}_{w\in\mathcal{A}^*}$, where p_w is the probability that an infinite string produced by the source on the denumerable alphabet $\mathcal{A}$ begins with w. P. Flajolet's methodology was then to put stochastic properties of the source into a correspondence with analytic properties of the Dirichlet series

$$\Lambda(s) = \sum_{w\in\mathcal{A}^*} p_w^s,$$

for a complex parameter s; thus a connection is forged between a stochastic and analytic approach, which yields results for analysis of algorithms. This unifying view illuminates, for instance, ways that the fundamental characteristics of the source (like the entropy) appear in critical ways during the analyses of algorithms or data structures related to the digital trees, especially when computing constants in asymptotic expansions related to generating functions of tree parameters. This approach allows one to relate, for example, the study of the famous Quicksort algorithm for strings [PF202] (considering keys are strings produced by a source) to the one of ternary search tries (which mixes binary search trees *inside* tries, whereas Quicksort for strings mixes tries inside a binary search tree).

A recurring theme in the analysis of tries is the appearance of minute oscillatory phenomena under some conditions. These oscillations are small but nonetheless are of great interest for the precise asymptotic mathematical analysis. In fact, this is inherent to the partitioning process, especially in the simplest stochastic model, in which words are drawn from an unbiased memoryless source. One key aspect for evaluating these oscillations together with error terms is to study precisely poles of the Mellin transform, relying on geometric and arithmetic conditions [PF208] of the source [6].

Methodology. We concentrate here more on tries. Some of the main parameters for tries are the size (the number of internal nodes) in relation with the memory space needed to store the data structure, the external path length (sum of the lengths of all paths from the root to all leaves) which relates to the construction cost of the data structure, and the height of trie (corresponding to the worst case for the number of

5. A study initiated by Devroye in [Dev84] for binary tries.

6. This leads to a surprising irruption of the Riemann hypothesis, to the great pleasure of P. Flajolet, when studying tries built upon words resulting from the continued fraction expansion of random numbers of the unit interval (see [PF161]).

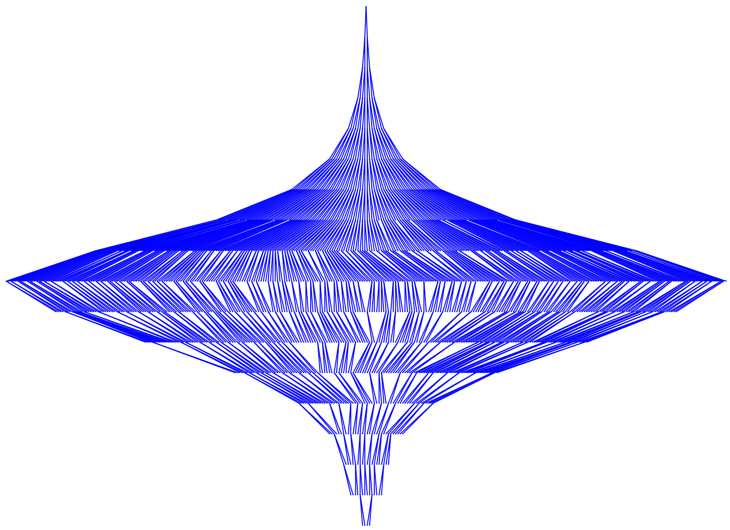

Figure 0.2. "A random trie of size n = 500 built over uniform data" [PF187]

symbol comparisons between any two strings in a set of strings). A random trie built over 500 uniformly generated strings, originally displayed in [PF187], is given in Figure 2.

As already mentioned, the studies differ whether we are interested in additive (size, path length) or multiplicative (height) parameters. However there are always two main steps: one is algebraic and provides exact expressions, the second one is analytic and aims at providing precise asymptotics.

An additive parameter γ for a trie $\mathcal{T}(Y)$ built on the set of strings Y is decomposed thanks to a "toll" function τ in the following recursive way

$$\gamma(\mathcal{T}(Y)) = \tau(Y) + \sum_{\alpha \in \mathcal{A}} \gamma(\mathcal{T}(Y \setminus \alpha)),$$

where $\tau(Y)$ is a toll function associated to the root of the trie. For instance, the toll functions for size (number of internal nodes), external path length are respectively (using the Iverson notation, i.e., $[\![B]\!]$ is one if property B is true and zero otherwise)

$$\tau(Y) = [\![\mathrm{Card}(Y) \geq 2]\!], \qquad \tau(Y) = \mathrm{Card}(Y) \times [\![\mathrm{Card}(Y) \geq 2]\!].$$

Concerning the algebraic step, when the number of strings in the trie is a Poisson random variable N with parameter z

$$\Pr(N = k) = e^{-z} \frac{z^k}{k!}.$$

(this is called a Poissonized model) instead of a fixed number n (in the traditional Bernoulli model), the calculations are greatly simplified in the resulting model. The Poisson distribution is well concentrated around the mean z, so that—by choosing the parameter $z = n$—the two models agree in many ways. Expressions in the Poisson model are also called Poissonised generating functions and are related to exponential generating functions. Then at this stage, for additive parameters of the trie, we usually obtain a functional equation for the Poissonised generating functions which can be iterated, yielding an infinite sum.

The variable z is then interpreted as a complex-valued variable, and asymptotic methods are used. The expressions computed at the previous stage are either directly or very close to harmonic sums of the form

$$G(z) = \sum_{k \in K} \lambda_k g(\mu_k z),$$

where families (λ_k) and (μ_k) are called amplitude and frequencies and g is the base function. Then the tool of choice to study asymptotics of harmonic sums is the Mellin transform as it isolates the transform of the base function and a Dirichlet-type series involving only the λ_k's and μ_k's. We refer to the corresponding chapter [FIXTHIS-CITE-Dumas-chapter] for a more precise description. The next step is then to relate the asymptotic development to the poles of the Mellin transform using methods from analytic combinatorics. The final, required step to perform depoissonisation, i.e., to interpret the degree to which the results from the Poissonized model are also valid in the (original) Bernoulli model, in which the number of strings is fixed.

We note that this is the most standard path to analyse tries (or related structures). This is sometimes referred as the Poisson-Mellin-Newton cycle. But there are other paths possible for instance using the Rice-Nörlund formula [PF202] (instead of Mellin transform) after an algebraic depoissonisation step.

The schema does not apply to multiplicative parameters. The precise analysis of the distribution of the height relies on a saddle point estimate [PF037].

Tries and the Analytic Combinatorics book. Although digital trees have been a focus of interest throughout the whole Philippe Flajolet's career, it is worth noting that digital trees are not present [7] in the book co-authored with R. Sedgewick [PF201] which is undoubtedly bound to be the reference in the field of analytic combinatorics. It may be useful to put this on perspective.

Indeed there was discussion between the authors, P. Flajolet and R. Sedgewick, to decide the fate of digital trees together with Mellin transform with respect to the book. The Mellin transform, and the analysis of tries as an advanced application, were in fact originally planned to be included as a book chapter [PF129]. These topics were omitted however, in the final stages, according to R. Sedgewick because revising and incorporating the full discussion would caused too much delay for this long awaited book. One may also infer that digital trees do not completely fit into the "philosophy" of the book, since problems related to a trie are of a more stochastic than combinatorial nature. Indeed, for digital trees, one is often confronted with generating functions for parameters satisfying a functional equation, from which an explicit expression (under

7. However we remark that tries are the subject of a chapter in the introductory book by the same authors Flajolet and Sedgewick [PF130] (with an analysis on the average relying only on elementary computations).

the form of an infinite sum) is deduced by iteration. But these generating functions are not generating functions for combinatorial structures.

3. Conclusion

This whole chapter illustrates why the digital tree process is central in computer science, and how the related analyses make intervene deep mathematical tools.

A small lecture guide. The beautiful survey [PF187] made by Philippe Flajolet himself is surely a wonderful entry point into the subject. The reader interested in algebraic methods for tries should be delighted with [PF053]. Two articles are more concerned with methodology and in particular extend analyses to digital search trees [PF101; PF061] (for digital search trees, functional equations now involve differentiation). Finally a general framework where strings are generated by very general sources—together with analyses of related structures (like tries) and sorting & searching algorithms—is found in [PF161; PF202; PF208].

Bibliography

[BCN12] F. Bassino, J. Clément, and P. Nicodème. "Counting Occurrences for a Finite Set of Words: Combinatorial Methods". In: *ACM Transactions on Algorithms* (2012+). Accepted for Publication.

[BD08] Nicolas Broutin and Luc Devroye. "An Analysis of the Height of Tries with Random Weights on the Edges". In: *Combinatorics, Probability & Computing* 17.2 (2008), pp. 161–202.

[Bri59] René de la Briandais. "File searching using variable length keys". In: *Papers presented at the the March 3-5, 1959, Western Joint Computer Conference*. IRE-AIEE-ACM '59 (Western). San Francisco, California, 1959, pp. 295–298.

[Dev84] L. Devroye. "A probabilistic analysis of the height of tries and of the complexity of triesort". In: *Acta Informatica* 21 (1984), pp. 229–237.

[Fay04] Julien Fayolle. "An average-case analysis of basic parameters of the suffix tree". In: *Mathematics and Computer Science III: Algorithms, Trees, Combinatorics and Probabilities*. Ed. by Michael Drmota et al. Trends in Mathematics. Proceedings of the 3rd International Colloquium of Mathematics and Computer Science, held at Technische Universität Wien. Basel: Birkhäuser Verlag, 2004, pp. 217–227.

[Fre60] Edward Fredkin. "Trie memory". In: *Communications of the ACM* 3 (1960), pp. 490–499.

[JS94] P. Jacquet and W. Szpankowski. "Autocorrelation on words and its applications. Analysis of suffix trees by string-ruler approach". In: *Journal of Combinatorial Theory* A66 (1994), pp. 237–269.

[KMW11] R. Kalpathy, H. M. Mahmoud, and M. D. Ward. *Asymptotic Properties of a Leader Election Algorithm*. 2011.

[Knu98] D. E. Knuth. *The Art of Computer Programming*. 2nd. Vol. 3. Reading, MA: Addison-Wesley, 1998.

[LSW07] S. Lonardi, W. Szpankowski, and M. D. Ward. "Error resilient LZ'77 data compression: algorithms, analysis, and experiments". In: *IEEE Transactions on Information Theory* 53 (2007), pp. 1799–1813.

[Mor68] Donald R. Morrison. "PATRICIA-Practical Algorithm To Retrieve Information Coded in Alphanumeric". In: *JACM* 15.4 (1968), pp. 514–534.

[PF034] Philippe Flajolet and Dominique Sotteau. "A recursive partitioning process of computer science". In: *Proceedings of the Second World Conference on Mathematics at the Service of Man*. Ed. by Antoni Ballester, David Cardús, and Enric Trillas. Las Palmas, Canary Islands, Spain: Universidad Politécnica de Las Palmas, 1982, pp. 25–30.

[PF036] Philippe Flajolet and Jean-Marc Steyaert. "A branching process arising in dynamic hashing, trie searching and polynomial factorization".

In: *Proceedings of the 9th International Colloquium on Automata, Languages and Programming (ICALP 82)*. Ed. by Mogens Nielsen and Erik Meineche Schmidt. Vol. 140. Lecture Notes in Computer Science. Aarhus, Denmark: Springer, 1982, pp. 239–251.

[PF037] Philippe Flajolet. “On the performance evaluation of extendible hashing and trie searching”. In: *Acta Informatica* 20 (1983), pp. 345–369.

[PF039] Philippe Flajolet. “Methods in the analysis of algorithms: Evaluations of a recursive partitioning process”. In: *Proceedings of the 1983 International Conference on Foundations of Computation Theory (FCT ’83)*. Ed. by Marek Karpinski. Vol. 158. Lecture Notes in Computer Science. Borgholm, Sweden: Springer, 1983, pp. 141–158.

[PF042] Philippe Flajolet and Nasser Saheb. “Digital search trees and the generation of an exponentially distributed variate”. In: *Proceedings of the 8th Colloquium on Trees in Algebra and Programming (CAAP ’83)*. Ed. by Giorgio Ausiello and Marco Protasi. Vol. 159. Lecture Notes in Computer Science. This version is subsumed by [PF060]. L’Aquila, Italy: Springer, 1983, pp. 221–235.

[PF050] Philippe Flajolet and G. Nigel Martin. “Probabilistic counting algorithms for data base applications”. In: *Journal of Computer and System Sciences* 31 (1985), pp. 182–209.

[PF053] Philippe Flajolet, Mireille Régnier, and Dominique Sotteau. “Algebraic Methods for Trie Statistics”. In: *Analysis and Design of Algorithms for Combinatorial Problems*. Ed. by Giorgio Ausiello and Mario Lucertini. Vol. 25. Annals of Discrete Mathematics. Papers based on the workshop held at the International Centre for Mechanical Sciences (CISM) in Udine, Italy. Equivalently cited as *North-Holland Mathematics Studies*, volume 109. North-Holland, 1985, pp. 145–188.

[PF060] Philippe Flajolet and Nasser Saheb. “The complexity of generating an exponentially distributed variate”. In: *Journal of Algorithms* 7 (1986). For preliminary versions, see [PF042]., pp. 463–488.

[PF061] Philippe Flajolet and Robert Sedgewick. “Digital Search Trees Revisited”. In: *SIAM Journal on Computing* 15 (1986), pp. 748–767.

[PF065] Philippe Flajolet and Philippe Jacquet. “Analytic models for tree communication protocols”. In: *Flow Control of Congested Networks*. Ed. by Amedeo R. Odoni, Lucio Bianco, and Giorgio Szegö. Vol. 38. NATO Advance Science Institute Series, *Series F: Computer and Systems Sciences*. Berlin/Heidelberg: Springer-Verlag, 1987, pp. 223–234.

[PF101] Philippe Flajolet and Bruce Richmond. “Generalized Digital Trees and Their Difference-Differential Equations”. In: *Random Structures & Algorithms* 3 (1992), pp. 305–320.

[PF120] Philippe Flajolet, Xavier Gourdon, and Philippe Dumas. “Mellin transforms and asymptotics: Harmonic sums”. In: *Theoretical Computer Science* 144 (1995), pp. 3–58.

[PF129] Philippe Flajolet and Robert Sedgewick. *The Average Case Analysis of Algorithms: Mellin Transform Asymptotics*. Research Report 2956. 93 pages. Institut National de Recherche en Informatique et en Automatique (INRIA), 1996.

[PF130] Robert Sedgewick and Philippe Flajolet. *An Introduction to the Analysis of Algorithms*. 512 pages. Reading, Massachusetts: Addison-Wesley Publishing Company, 1996.

[PF140] Julien Clément, Philippe Flajolet, and Brigitte Vallée. "The Analysis of Hybrid Trie Structures". In: *Proceedings of the Ninth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '98)*. San Francisco, CA, USA, 1998, pp. 531–539.

[PF157] Philippe Flajolet and Brigitte Vallée. "Continued Fractions, Comparison Algorithms, and Fine Structure Constants". In: *Constructive, experimental, and nonlinear analysis*. Ed. by Michel A. Théra. Vol. 27. Conference Proceedings, Canadian Mathematical Society. Providence, RI, USA: American Mathematical Society, 2000, pp. 53–82.

[PF161] Julien Clément, Philippe Flajolet, and Brigitte Vallée. "Dynamical sources in information theory: A general analysis of trie structures". In: *Algorithmica* 29 (2001), pp. 307–369.

[PF187] Philippe Flajolet. "The Ubiquitous Digital Tree". In: *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science (STACS 2006)*. Ed. by Bruno Durand and Wolfgang Thomas. Vol. 3884. Lecture Notes in Computer Science. Marseille, France: Springer, 2006, pp. 1–22.

[PF200] Philippe Flajolet, Maryse Pelletier, and Michèle Soria. "On Buffon Machines and Numbers". In: *Proceedings of the Twenty-Second Annual ACM–SIAM Symposium on Discrete Algorithms (SODA '11)*. San Francisco, CA, USA, 2011, pp. 172–183.

[PF201] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. xiv+810 pages. Also available electronically from the authors' home pages. Cambridge University Press, 2009.

[PF202] Brigitte Vallée et al. "The Number of Symbol Comparisons in QuickSort and QuickSelect". In: *Proceedings of the 36th International Colloquium on Automata, Languages and Programming: Part I (ICALP 2009)*. Ed. by Susanne Albers et al. Vol. 5555. Lecture Notes in Computer Science. Rhodes, Greece: Springer, 2009, pp. 750–763.

[PF208] Philippe Flajolet, Mathieu Roux, and Brigitte Vallée. "Digital Trees and Memoryless Sources: from Arithmetics to Analysis". In: *21st International Meeting on Probabilistic, Combinatorial, and Asymptotic Methods in the Analysis of Algorithms (AofA '10)*. Ed. by Michael Drmota and Bernhard Gittenberger. Vol. AM. DMTCS Proceedings. 2010, pp. 233–260.

[Pit85] Boris Pittel. "Asymptotical Growth of a Class of Random Trees". In: *Annals of Probability* 13 (1985), pp. 414–427.

[Pro93] H. Prodinger. "How to select a loser". In: *Discrete Mathematics* 120 (1993), pp. 149–159.

[Thu12] Axel Thue. "Skrifter udgivne of Videnskabs-Selskabet i Christiania". In: *Mathematisk-Naturvidenskabelig Klasse* (1912). Reprinted in Thue's Selected Mathematical Papers (Oslo: Universitetsforlaget, 1977), pages 413–477.

[ZL77] Jacob Ziv and Abraham Lempel. "A universal algorithm for sequential data compression". In: *IEEE Transactions on Information Theory* IT-23 (1977), pp. 337–343.