1991

How to Count Quickly and Accurately: A Unified Analysis of Probabilistic Counting and Other Related Problems

Peter Kirschenhofer

Helmut Prodinger

Wojciech Szpankowski
Purdue University, spa@cs.purdue.edu

Report Number:
91-079

Kirschenhofer, Peter; Prodinger, Helmut; and Szpankowski, Wojciech, "How to Count Quickly and Accurately: A Unified Analysis of Probabilistic Counting and Other Related Problems" (1991). *Department of Computer Science Technical Reports*. Paper 918.
<https://docs.lib.purdue.edu/cstech/918>

**HOW TO COUNT QUICKLY AND ACCURATELY:
A Unified Analysis of Probabilistic
Counting and Other Related Problems**

**Peter Kirschenhofer
Helmut Prodinger
Wojciech Szpankowski**

**CSD-TR-91-079
November 1991**

HOW TO COUNT QUICKLY AND ACCURATELY: A Unified Analysis of Probabilistic Counting and Other Related Problems

Peter Kirschenhofer*

Dept. of Algebra&Discrete Math.
Technical University of Vienna
Vienna A-1040
Austria

Helmut Prodinger*

Dept. of Algebra&Discrete Math.
Technical University of Vienna
Vienna A-1040
Austria

Wojciech Szpankowski[†]

Dept. of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.

Abstract

We consider a class of probabilistic counting algorithms parameterized by an integer $d \geq 0$ that estimate the number of distinct elements N in a large set. Our algorithms generalize the idea of Flajolet and Martin [2], who limited themselves to the case $d = 0$. We construct an unbiased estimator of N that is *asymptotically optimal* ($d \rightarrow \infty$) in the sense that the variance of the estimator is as small as possible. The variance becomes $1/12 = 0.083$ as $d \rightarrow \infty$ that is the best possible in a class of counting algorithms investigated in this paper since the correction due to discretization is equal to $1/12$ (so called *Sheppard's correction*), and cannot be beaten by any algorithm within this class. This can be compared to the variance 1.257 of Flajolet and Martin's algorithm. As noted by Brassard and Bratley [1] "it is far from obvious how to carry out a more precise analysis of the unbiased estimate of $N \dots$ ". We present a novel and complete analysis of these new counting algorithms that – to the best of our knowledge – cannot be obtained by an extension of the analysis in [2]. We present results concerning the average value, the variance, the limiting distribution and the limiting generating function of an estimate of N . Moreover, our novel approach is *not* limited to probabilistic counting algorithms, and it can be applied in the investigation of several other "splitting algorithms" such as selecting the loser [15], estimating the number of questions necessary to identify the number of distinct objects [14], searching algorithms based on digital tries, approximate counting, electing d finalists in a contest (cf. polling system), and so forth.

1. INTRODUCTION

How many distinct words are in the *Webster Dictionary* (e.g., stored in a computer memory of your SUN-Workstations)? How many distinct molecular sequences are stored in the GeneBank databases? In several algorithms on databases a major determinant of efficiency is the *cardinality* of the underlying set. For example, the computation of the intersection of two sets may be drastically speed up if one eliminates duplicates in both sets, and estimates cardinalities of these sets. This is even more important for some operations on a relational database. The interested reader is referred to Flajolet and Martin [2] (cf. also [1]) for more motivations of probabilistic counting.

*This work was supported by Fonds zur Förderung der Wissenschaftlichen Forschung Project P7497-TEC

[†]This research was supported by the NSF Grant INT-8912631, and in part by AFOSR grant 90-0107, by the NSF grant CCR-8900305, and by grant R01 LM05118 from the National Library of Medicine.

In order to make these observations practical, one needs an efficient way of estimating the cardinality of large (multi)sets of data. Clearly, the trivial method of building a list of elements without replications is unacceptable due to the cost of disk access and auxiliary memory. Knowing this fact, Flajolet and Martin [2] proposed an algorithm that is probabilistic in its nature. It works as follows. In order to estimate the cardinality N of a set $\mathcal{M}$ (with replications) every element $x \in \mathcal{M}$ is hashed into a binary string of size m (the choice of m is easy, and $m = 5 + \log N$ suffices [1]). The bitwise OR-composition of modified hashed strings is used to build the so called *bitmap* and to obtain the estimate R_N of the cardinality N of $\mathcal{M}$. More precisely, the position of the leftmost zero in the *bitmap* string approximates $\log_2 N$ (for details see below and [2], [1]).

A precise estimate of R_N “*is far from obvious how to carry out*” [1]. Such an analysis is crucial to obtain a precise estimate of N , and of prime importance to build an estimator of N that is *unbiased* with the variance as small as possible. More importantly, one must assure a small deviation in the distribution of the estimator around its mean value (i.e., *limiting distribution* of $R_N - ER_N$). Flajolet and Martin in their seminal paper [2] presented a very careful analysis of R_N . Unfortunately, the variance of the estimator R_N turned out to be large and equal approximately to 1.257. Brassard and Bratley write “.. *the standard deviation of R_N shows that this estimate may be in error by a factor of 2, which is unacceptable*”. A modification of the basic algorithm was proposed also in [2] but this new algorithm is more complex and more costly.

In our paper, we use the idea of Flajolet and Martin to propose a new estimate $R_{N,d}$ of N that is parameterized by an integer d . The value $d = 0$ corresponds to Flajolet-Martin’s basic algorithm. Our algorithm has the best possible variance equal to $1/12 = 0.083$ as $d \rightarrow \infty$. The new idea behind our algorithm is to *count* the number of 1’s up to $d + 1$ in every position of the *bitmap*. As an estimate of N we propose to take the leftmost position that contains the value smaller than $d + 1$. More formally, our algorithm works as follows. We assume that there exists a carefully implemented hashing function $hash : \mathcal{M} \rightarrow \{0, 1\}^m$ where m is chosen as in [2]. Let y be the image of $x \in \mathcal{M}$ over the $hash(\cdot)$ function, that is, $y = hash(x)$. The procedure below constructs another string called *bitmap* that is used to estimate N . As in [2], we define the function $\rho(y)$ for a binary string y as the leftmost position that contains value 1.

procedure *BITMAP*

for $i = 0$ **to** $m - 1$ **do** *bitmap*(i) = 0 ;

for $x \in \mathcal{M}$ **do**

begin

$y = hash(x)$;

for $i = 0$ **to** $\rho(x)$ **do**

$bitmap(i) = \min\{bitmap(i) + y(i), d + 1\}$;

end;

end;

Note that the resulting string *bitmap* is a $d + 1$ -ary string. For example, for $d = 3$ we may have *bitmap* = 444444444434232130. As the estimate $R_{N,d}$ we take the position of the leftmost value smaller than $d + 1$ (e.g., in the example above we have $R_{N,3} = 11$). More precisely we define:

$$R_{N,d} = \min\{k : \text{bitmap}(k) < d + 1 \text{ and for all } 0 \leq i < k \text{ bitmap}(i) = d + 1\} . \quad (1)$$

It is easy to see how our algorithm works. The *bitmap* string is a sequence of integers from the set $\{0, 1, \dots, d + 1\}$. If $\text{bitmap}(i) = k < d + 1$, then exactly k strings y have at the i th position the leftmost one. Moreover, $\text{bitmap}(i) = d + 1$ means that *at least* $d + 1$ strings y contain the leftmost one at position i .

We now summarize our main results concerning the probabilistic behavior of $R_{N,d}$. In particular, we prove that $ER_{N,d} = \log_2 N - 1 - C_d + P(\log_2 N) + O(\sqrt{N})$ where C_d is a constant such that $C_d \rightarrow \log_2 d - 0.5 + O(d^{-1/2+\epsilon})$ as $d \rightarrow \infty$, and $P(x)$ is a periodic function of period 1 with a small amplitude. More importantly, we show that the variance of $R_{N,d}$ is very small, namely $\text{var} R_{N,d} \rightarrow 1/12 = 0.08333 \dots$ as $d \rightarrow \infty$ which is the best possible value for the variance within the class of algorithms investigated in this paper. This last assertion is a direct consequence of the so called Sheppard's correction (cf. [9]) which says that the discretization by itself introduces some fluctuations with the variance equal to $1/12$. Finally, in the next section we also obtain the limiting distribution and the limiting generating function for $R_{N,d}$, that is, we assess the deviation of $R_{N,d}$ around its mean value.

Literature on the probabilistic counting is rather scanty. It was initiated by the seminal paper of Flajolet and Martin [2] who analyzed the case $d = 0$ of our algorithm. Their analysis is based on an application of the inclusion-exclusion rule, and – to the best of our knowledge – cannot be extended to $d > 0$. This is also evident from the analysis of Greenberg, Flajolet and Ladner [6] who analyzed a conflict resolution algorithm which resembles the case $d = 1$ of our analysis. A simplification of Flajolet-Martin's analysis is presented in Kirschenhofer and Prodinger [11]. Finally, a different estimator for the probabilistic counting algorithm was analyzed in Szpankowski and Rego [18] who investigated the maximum over N geometrically distributed random variables. The authors of [18] considered the *last* bit that is equal to one in the *bitmap*. Unfortunately, in this case the variance is even bigger and equal to 3.507.

It must be stressed that the novelty of our paper also lies in the area of the mathematical analysis of algorithms. In fact, our technique of analysis can be applied to several other algorithms. As shown in section below, we reduce our probabilistic counting algorithm to a problem on (*digital*) *trees*. Evaluating a parameter of such a tree, we obtain the following functional equation

$$g(z) = \beta a(z/2)g(z/2) + b(z) . \quad (2)$$

In general, this equation does not have an explicit finite solution unless $a(z)$ is a very particular function. For example, $a(z) = e^z$ was discussed in Knuth [12] (cf. also [17]), and

Szpankowski [16], [17] (cf. also [12]) investigated the case $a(z) = e^z + 1$. For these two cases explicit final solutions of (2) were obtained, but it is rather hopeless to expect the existence of a useful closed form solution for general $a(z)$ and $b(z)$. An asymptotic solution to (2) was first suggested in Jacquet and Szpankowski [8] who analyzed the case $a(z) = 1 + (1+z)e^{-z}$. Recently, Flajolet and Richmond [4] investigated asymptotically the above equation with $a(z) = 2z^b/(1+z)^b$ for some integer $b > 0$. In the analysis of the probabilistic counting algorithms introduced in this paper, we must investigate (2) with $a(z) = 1 - e_d(z)e^{-z}$ where $e_d(z) = \sum_{i=0}^d z^i/i!$. In fact, in the extended version of this paper we intend to present a general approach that can be used to derive an asymptotic solution for a large class of functional equation of type (2).

2. MAIN RESULTS

In this section we present our main results. We start with a short description of the probabilistic counting algorithm. Our description below will slightly differ from the one presented in Section 1, but it helps to establish recurrence equations necessary for the analysis.

Our new description goes as follows. Let us consider an empty *bitmap* string, that is, with all positions filled by zeros. Assume that N objects (e.g., data, persons, etc.) can randomly insert (hit) a 1 at any position of the *bitmap*, however, probability of hitting the $j \geq 0$ position is equal to 2^{-j-1} . In terms of the probabilistic counting, this means that the probability of the occurrence of the pattern like $0^j 1 \dots$ is equal to 2^{-j-1} since 0 and 1 are equally likely (0^j denotes j consecutive zeros). Every object can hit only one time. In addition, we count the number of hits in any position of the *bitmap*, but we count the number of hits only up to some value $d+1$, where d is a given parameter. In other words, the *bitmap* is a $d+1$ -ary string. Clearly, the *bitmap* will contain a lot of $d+1$'s at the beginning of the string. This is due to the fact that the probability of a hit decreases exponentially fast with the increase of the position in the *bitmap*. The parameter of interest is the length $R_{N,d}$ of the longest run of $d+1$'s in the front of the *bitmap*. A precise definition of $R_{N,d}$ is given in (1).

Let $F_N(z) = Ez^{R_{N,d}}$ be the probability generating function of the estimator $R_{N,d}$. That is, $[z^j]F_N(z)$ (the j th coefficient at $F_N(z)$) is the probability that the *bitmap* is $(d+1)^j c \dots$ where $c < d$, and $(d+1)^j$ denotes j consecutive $(d+1)$ values. To derive the recurrence for $F_N(z)$, one may represent the action of our algorithm as a splitting process in a digital tree. Imagine N persons flipping a fair coin, and those who get 1 (a hit) move to the right while the rest to the left. The process continues until less than $d+1$ persons flip a 1. Imagine that at the first step k persons flip 1. Then, the contribution to the generating function $F_N(z)$ is as follows: if $k \leq d$ then we need to add $z^0 \sum_{k \leq d} \binom{N}{k} 2^{-N}$, otherwise the contribution becomes $z \sum_{d+1 \leq k \leq N} \binom{N}{k} 2^{-N} F_{N-k}(z)$. In short, we obtain the following recurrence

$$F_N(z) = \sum_{k=0}^d \binom{N}{k} 2^{-N} + z \sum_{k=0}^{N-d-1} \binom{N}{k} 2^{-N} F_k(z), \quad (3)$$

which is valid for all $N \geq 0$.

Using (3) we can derive recurrences for the first and second moments of $R_{N,d}$, namely: $ER_{N,d} = F'_N(1)$ and $\text{var } R_{N,d} = F''_N(1) + F'_N(1) - (F'_N(1))^2$. In particular, let $L(z)$ and $W(z)$ be the exponential generating function for $F'_N(1)$ and $F''_N(1)$ respectively. That is, $L(z) = \sum_{N \geq 0} F'_N(1) z^N / N!$ and $W(z) = \sum_{N \geq 0} F''_N(1) z^N / N!$. Define also $\tilde{L}(z) = e^{-z} L(z)$ and $\tilde{W}(z) = e^{-z} W(z)$. Then, (3) implies

$$\tilde{L}(z) = f_d(z/2) \tilde{L}(z/2) + f_d(z/2) \quad (4)$$

$$\tilde{W}(z) = f_d(z/2) \tilde{W}(z/2) + 2 \tilde{L}(z) f_d(z/2) \quad (5)$$

where $f_d(z) = 1 - e_d(z)e^{-z}$ and $e_d(z) = 1 + z^1/1! + \dots + z^d/d!$.

Both functional equations fall into the general equation (2) which we repeat below

$$g(z) = \beta a(z/2) g(z/2) + b(z) . \quad (6)$$

where $\beta \leq 1$. Iterating (6), we obtain a general solution in the form of (cf. [8])

$$g(z) = \sum_{n=0}^{\infty} \beta^n b(z 2^{-n}) \prod_{k=1}^n a(z 2^{-k}) \quad (7)$$

if $g(0) = 0$. Define

$$\varphi(z) = \prod_{j=0}^{\infty} a(z 2^j) , \quad (8)$$

provided the infinite product in (8) converges. Then, the general solution (7) can be rewritten as

$$g(z) \varphi(z) = \sum_{n=0}^{\infty} \beta^n b(z 2^{-n}) \varphi(z 2^{-n}) . \quad (9)$$

The last form is the clue to our asymptotic solution of the functional equation (7), and in particular to solve (4) and (5). In Section 3 we present our detailed derivations.

Now we are ready to present our main results.

Theorem 1. *Consider the probabilistic counting algorithms described above, and in particular the quantity $R_{N,d}$.*

(i) *The average value of $R_{N,d}$ becomes asymptotically as $N \rightarrow \infty$*

$$ER_{N,d} \sim \log_2 N - 1 - C_d + P_1(\log_2 N) + O(1/\sqrt{N}) \quad (10)$$

where $P_1(x)$ is a periodic function with period 1 and a very small amplitude, and with $L = \log 2$ we have

$$\begin{aligned} C_d &= \frac{1}{L^2} \int_0^{\infty} e^{-x} e_d(x) \varphi(2x) \frac{\log x}{x} dx \\ &\sim \log_2 d - 0.5 + O(d^{-1/2+\varepsilon}) \quad \text{as } d \rightarrow \infty . \end{aligned} \quad (11)$$

(ii) The variance of $R_{N,d}$ is for $N \rightarrow \infty$

$$\text{var} R_{N,d} \sim D_d - C_d^2 + P_2(\log_2 N) + O(1/\sqrt{N}),$$

where

$$D_d - C_d^2 \sim \frac{1}{12} + O(d^{-1/2+\epsilon}) \quad \text{as } d \rightarrow \infty. \quad (12)$$

In particular

$$\begin{aligned} D_d &= \frac{1}{L^3} \int_0^\infty e_d(x) e^{-x} \varphi(2x) \frac{\log^2 x}{x} dx \\ &\sim -C_d - 1/6 + \log_2^2 d + O(d^{-1/2+\epsilon}) \quad \text{as } d \rightarrow \infty, \end{aligned} \quad (13)$$

and $P_2(x)$ is a periodic function with period 1 and a small amplitude. ■

Using numerical integration, we computed for the first few values of d the constant $ER_{N,d} - \log_2 N - P_1(\log_2 N)$ and $\text{var} R_{N,d} - P_2(\log_2 N)$. They are displayed in the table below. We note that the variance decreases substantially for initial values of d ($d \leq 2$) and then the decrease slows down. From the table, one may conclude that $d \geq 5$ is a good practical choice.

Table 1: Performance of the algorithm for small values of d .

d	$ER_{N,d} - \log_2 N$	$\text{var} R_{N,d}$
0	-0.37	1.26
1	-1.40	0.78
2	-2.00	0.59
3	-2.42	0.49
4	-2.74	0.43
5	-3.02	0.38
6	-3.24	0.35
7	-3.43	0.32

Our technique also allows to derive the limiting distribution for $R_{N,d}$ and the limiting generating function. To see this, we introduce the Poisson transform of $F_N(z)$ as

$$G(z, x) = e^{-x} \sum_{N=0}^{\infty} F_N(z) \frac{x^N}{N!}. \quad (14)$$

Then, after some algebra, one proves that $H(z, x) = G(z, x) - 1$ satisfies the following recurrence

$$H(z, x) = 2f_d(x/2)H(z, x/2) + (z - 1)f_d(x/2), \quad (15)$$

which falls into our general functional equation (6). In particular, as above we can write a solution for $H(z, x)$ as

$$\varphi(x)H(z, x) = (z - 1) \sum_{n=0}^{\infty} z^n \varphi(x2^{-n-1}), \quad (16)$$

where $\varphi(x)$ is defined in (8).

With the help of (16), we are able to derive the limiting generating function for $F_N(z)$ as N becomes large. In addition, we can obtain the limiting distribution for $R_{N,d}$. We summarize these results in the theorem below.

Theorem 2. *Consider the probabilistic counting algorithm as in Theorem 1.*

(i) *For large N the limiting generating function becomes*

$$F_N(z) \sim z^{\log_2 N} \varphi^*(-\log_2 z)/L + P_3(\log_2 N) + O(1/\sqrt{N}). \quad (17)$$

where $\varphi^*(s)$ is the Mellin transform of $\varphi(x)$ defined in (8).

(ii) *The limiting distribution of $R_{N,d}$ can be written as*

$$\lim_{N \rightarrow \infty} \Pr\{R_{N,d} - \log_2 N \leq x\} \sim \frac{1}{\varphi(2^{-x})}. \quad (18)$$

where $\varphi(x)$ is defined in (8). ■

We point out that the limiting generating function (17) seems to be more convenient for the derivation of moments. The limiting distribution (18) provides information about the concentration of $R_{N,d}$ around its mean value.

Remarks

(i) *Special Cases.* As mentioned before, the case $d = 0$ was analyzed by Flajolet and Martin [2]. In addition, Greenberg *et al.* [6] investigated the function $\varphi(x)$ for $d = 1$ in the analysis of a conflict resolution algorithm for multiple channels. It must be stressed that the analysis of Flajolet and Martin [2] is based on the inclusion-exclusion rule, and seems to be very unlikely extendable to $d \geq 1$. This is partially evident from the analysis in [6]. We note that our analysis is based on a quite different approach, namely, functional equations, and can be proved very successful for some other algorithms (cf. Remark (iii)).

(ii) *Convergence in Probability.* Theorem 1 implies that $R_{N,d}$ converges *in probability* (pr.) to $ER_{N,d}$. More precisely, by Theorem 1 and Chebyshev's inequality we have

$$\Pr\{|R_{N,d}/ER_{N,d} - 1| > \varepsilon\} \leq \frac{\text{var } R_{N,d}}{\varepsilon^2 E^2 R_{N,d}} \sim \frac{\text{const}}{\log_2^2 N} \rightarrow 0$$

as $N \rightarrow \infty$. This proves the announced convergence. It is not clear, however, whether one can extend this convergence to the *almost sure* convergence. However, a more careful analysis of the limiting distribution (18) might lead to an answer.

(iii) *Other Problems.* As mentioned in the abstract, our technique can be applied to analyze several other algorithms that are based on the “splitting process” described at the beginning of this section. More precisely, a splitting algorithm divides randomly N objects according to some rule. A good representation of such a splitting process is a digital tree (i.e., a trie [12]). For example, consider a multiaccess system with a large number of users. The following “conflict resolution” algorithm found to be very successful for some multiaccess systems (cf. [6], [8]). Let N be the size of a conflict, that is, the number of users that simultaneously sent packets. To split it, every user flips a fair coin, and those who get 1 “move right” in a digital tree represented this process, while the others “move left”. This splitting process continues until all users sent successfully their packets (i.e., subtrees containing conflicting users are of size one). This algorithm can be analyzed in a unified manner by our approach through the functional equation of type (6) and its solution (9). Here is another example that generalizes PATRICIA tries (cf. [12], [17]). Consider N infinite strings that are built over a V -ary alphabet. Using a trie construction as in Knuth [12] (see also [17]) we can split all strings in such a manner that no two of them will share the common prefix. A trie built in such a manner may contain some internal nodes that are unary. To avoid such a waste of storage, one may compress this trie to obtain a PATRICIA trie. In the PATRICIA all internal nodes have degree greater equal to two. But a generalization of a PATRICIA trie might be also explored by imposing that all internal nodes have degree of size greater equal to $d \leq V$. What is the search time (i.e., the length of a path to an external node) in such a trie? What about the average height, etc.? The analysis of such a data structure seems to be a nontrivial task, and our approach can be applied to obtain some asymptotics. Finally, several other problems fall into the “splitting process” scenario. We mention here: selecting the loser [15], estimating the number of questions necessary to identify the number of distinct objects [14], selecting d finalist in a contest where questions are answered randomly, and so forth.

3. ANALYSIS

In this section we present a sketch of proof for Theorem 1. We also offer some remarks regarding Theorem 2.

We start with the average value $ER_{N,d}$. Its modified exponential generating function $\tilde{L}(z)$ satisfied the functional equation (4) which we repeat below

$$\tilde{L}(z) = f_d(z/2)\tilde{L}(z/2) + f_d(z/2) \quad (19)$$

where $f_d(z) = 1 - e_d(z)e^{-z}$ and $e_d(z) = 1 + z^1/1! + \dots + z^d/d!$. Iterating this equation, as in (6) – (9), we finally obtain

$$\tilde{L}(z)\varphi(z) = \sum_{n=0}^{\infty} \varphi(z2^{-n-1}) . \quad (20)$$

where

$$\varphi(z) = \prod_{j=0}^{\infty} f_d(z2^j) , \quad (21)$$

The idea of our analysis is to obtain an asymptotic behavior of $\tilde{L}(z)$ as $z \rightarrow \infty$, and then by *depoissonization* [8] to evaluate the coefficients $ER_{N,d}$ of $L(z)$. To obtain the asymptotics of $\tilde{L}(z)$ we apply the Mellin transform technique (cf. [5] and [3]). We shall use the following depoisonization lemma proved in Jacquet and Szpankowski [8].

Depoissonization Lemma. *Let $f(x) = e^{-x}F(x) = e^{-x} \sum_{n=0}^{\infty} f_n x^n / n!$ and let also define $S_\theta = \{x : |\arg x| < \theta, 0 < \theta < \pi/2\}$. If for $x \in S_\theta$ and $x \rightarrow \infty$*

$$f(x) \sim x^p \log^q x \quad (22)$$

for some p and q , and for $x \notin S_\theta$

$$|F(x)| < B e^{\alpha x} \quad (23)$$

for some $B > 0$ and $0 < \alpha < 1$, then the coefficient f_n of $F(x)$ becomes asymptotically

$$f_n = f(n) \left(1 + O\left(\frac{1}{\sqrt{n} \log n}\right) \right) \quad (24)$$

as $n \rightarrow \infty$. ■

Using the above idea, we first apply the Mellin transform to (19). Let $Q_1(z)$ and $Q_1^*(s)$ be respectively the right-hand side (in short: RHS) of (19) and its Mellin transform. Since the sum in (19) is a harmonic sum, hence (cf. [5])

$$Q_1^*(s) = \frac{2^s}{1-2^s} \varphi^*(s) = \left(\frac{2^s}{1-2^s} \right)^2 \Phi^*(s) \quad (25)$$

where $\varphi^*(s)$ is the Mellin transform of $\varphi(z)$, and $\varphi^*(s) = \Phi^*(s)/(2^{-s} - 1)$. Note that $\Phi^*(s)$ is an entire function (cf. [6]). Using the inverse of the Mellin transform (by computing the residues of $Q_1^*(s)$) we obtain for $z \rightarrow \infty$

$$\tilde{L}(z) \sim \log_2 z \cdot \frac{\Phi^*(0)}{L} - \frac{\Phi^*(0)}{L} - \frac{\Phi'^*(0)}{L^2} + P(z) \quad (26)$$

where

$$\begin{aligned} \Phi^*(0) &= \int_0^\infty e^{-x} e_d(x) \varphi(2x) \frac{dx}{x}, \\ \Phi'^*(0) &= \int_0^\infty e^{-x} e_d(x) \varphi(2x) \frac{\log x}{x} dx, \end{aligned}$$

and $P(z)$ is a fluctuating function with a small amplitude.

We need to evaluate $\Phi^*(0)$ and $\Phi'^*(0)$. The former is rather easy, and by properties of the Mellin transform we find out that $\Phi^*(0) = L = \log 2$. The latter quantity is harder to estimate. To do so, we first present another evaluation of $\Phi^*(0)$ which can be extended to obtain $\Phi'^*(0)$. Note that $\varphi(2x) - \varphi(x) = e^{-x} e_d(x) \varphi(2x)$. Define also a function $1(x)$ as follows

$$1(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ 0 & \text{if } x < 1. \end{cases}$$

Then, we can write

$$\begin{aligned}
\Phi^*(0) &= \int_0^\infty (\varphi(2x) - \varphi(x)) \frac{dx}{x} \\
&= \int_0^\infty (\varphi(2x) - 1(2x)) \frac{dx}{x} + \int_0^\infty (1(2x) - 1(x)) \frac{dx}{x} + \int_0^\infty (1(x) - \varphi(x)) \frac{dx}{x} \\
&= \int_0^\infty (1(2x) - 1(x)) \frac{dx}{x} = \int_{1/2}^\infty \frac{dx}{x} = L.
\end{aligned}$$

In a similar manner we can deal with $\Phi'^*(0)$. This time, however, the calculation is more involved. Nevertheless, we obtain the following

$$\begin{aligned}
\Phi'^*(0) &= -L^2/2 - L \int_0^\infty (1(x) - \varphi(x)) \frac{dx}{x} \\
&= \int_0^1 \varphi(x) \frac{dx}{x} + \int_1^\infty (\varphi(x) - 1) \frac{dx}{x} \\
&\sim O(d^{-1/2}) - \int_1^\infty e_d(x) e^{-x} \frac{dx}{x} \quad d \rightarrow \infty
\end{aligned}$$

where the last integral can be analyzed using the *incomplete gamma* function. In particular, we can prove that for $d \rightarrow \infty$ we have $C_d = \frac{1}{L^2} \Phi'^*(0) = \log_2 d - 1/2 + O(d^{-1/2+\epsilon})$. Finally, using the Depoissonization Lemma we establish Theorem 1(i).

The proof of Theorem 1(ii) concerning the variance of $R_{N,d}$ is more intricate but it proceeds along the same lines as the above derivation. In particular, we deal now with the functional equation (5) which we repeat below

$$\widetilde{W}(z) = f_d(z/2) \widetilde{W}(z/2) + 2\widetilde{L}(z) f_d(z/2). \quad (27)$$

Using the idea from Section 2 (cf. (6) – (9)) we find immediately an explicit solution to (27), namely

$$\widetilde{W}(z) \varphi(z) = 2 \sum_{n=0}^\infty \widetilde{L}(z 2^{-n-1}) \varphi(z 2^{-n-1}),$$

which becomes, after inserting the solution (20) for $\widetilde{L}(z)$,

$$\widetilde{W}(z) \varphi(z) = 2 \sum_{j=1}^\infty \sum_{n=1}^\infty \varphi(z 2^{-n-j}). \quad (28)$$

To establish asymptotics of $\widetilde{W}(z)$, and also $F_N''(1)$, we proceed as before through the Mellin transform and depoissonization. Let $Q_2(z)$ and $Q_2^*(s)$ denote respectively the RHS of (28) and its Mellin transform. Then,

$$Q_2^*(s) = 2 \left(\frac{2^s}{1-2^s} \right)^2 \varphi^*(s) = 2 \left(\frac{2^s}{1-2^s} \right)^3 \Phi^*(s)$$

where $\Phi^*(s)$ is defined as before. Computing residues of the above, one immediately proves for $z \rightarrow \infty$

$$Q_2(z) \sim \log_2^2 z - (3 + 2C_d) \log_2 z + 2 + 3C_d + D_d + P(\log_2 z)$$

where C_d was defined before, and

$$D_d = \frac{1}{L^3} \Phi^{*''}(0) = \frac{1}{L^3} \int_0^\infty e_d(x) e^{-x} \varphi(2x) \frac{\log^2 x}{x} dx .$$

We can analyze D_d in the same manner as before. After some algebra we arrive at

$$\begin{aligned} D_d &\sim C_d - 1/6 + 2/L^2 \int_1^\infty e_d(t) e^{-t} \frac{\log t}{t} dt \\ &\sim C_d - 1/6 + \log_2^2 d + O(d^{-1/2+\varepsilon}) \end{aligned}$$

as $d \rightarrow \infty$. After applying the Depoissonization Lemma, we complete the proof of Theorem 1(ii).

The limiting generating function $\lim_{N \rightarrow \infty} F_N(z)$ can be analyzed in the same manner as above but this time the solution (16) should be used to derive (17) of Theorem 2(i). For the limiting distribution we also used the solution (16). Note that the k th coefficient of $H(z, x)$ suffices to determine this distribution. From (16) we find that $[z^k](F_N(z)/(1-z)) = \varphi(N/2)/\varphi(N2^{-k}) + O(N^{-1/2})$. Substituting $k = \log_2 N + x$ and taking $N \rightarrow \infty$ will lead to our formula (18) in Theorem 2(ii). Details of these derivations are left to a journal version of the paper.

Acknowledgment. The first two authors want to thank M. Drmota and P. Grabner for interesting discussions.

References

- [1] G. Brassard and P. Bratley, *Algorithmics. Theory and Practice*, Prentice Hall, Englewood Cliffs, 1988.
- [2] P. Flajolet and G.N. Martin, Probabilistic Counting Algorithms for Data Base Applications, *J. Comp. Syst. Sci.*, 31, 182-209, 1985.
- [3] P. Flajolet and A. Odlyzko, Singularity Analysis of Generating Functions, *SIAM J. Disc. Methods*, 3, 216-240, 1990.
- [4] P. Flajolet and B. Richmond, Generalized Digital Trees and Their Difference-Differential Equations, INRIA TR-1423, 1991.
- [5] P. Flajolet, M. Régnier and R. Sedgewick, Some Uses of the Mellin Transform Techniques in the Analysis of Algorithms, in *Combinatorial Algorithms on Words*, Springer NATO ASI Ser. F12, 241-254, 1985.
- [6] A. Greenberg, P. Flajolet and R. Ladner, Estimating the Multiplicities of Conflicts to Speed Their Resolution in Multiple Access Channels, *JACM*, 34, 289-325, 1987.
- [7] P. Henrici, *Applied and Computational Complex Analysis*, vol. 2, John Wiley & Sons 1977.

- [8] P. Jacquet and W. Szpankowski, Ultimate Characterizations of the Burst Response of an Interval Searching Algorithm: A Study of a Functional Equations, *SIAM J. Computing*, 18, 777-791, 1989.
- [9] M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, 4th Ed. vol. 1, Charles and Griffin and Co. Ltd., London (1977).
- [10] P. Kirschenhofer and H. Prodinger, On the Analysis of Probabilistic Counting, *Lecture Notes in Mathematics*, 1452 (Eds. E. Hlawka and R. Tichy), 117-120, Springer Verlag 1990.
- [11] P. Kirschenhofer and H. Prodinger, Approximate Counting: An Alternative Approach, *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 25, 43-48, 1991.
- [12] D.E. Knuth, *The Art of Computer Programming. Sorting and Searching*, vol. 3., Addison-Wesley, Reading, MA 1973.
- [13] R. Morris, Counting Large Numbers of Events in Small Registers, *Comm. ACM*, 21, 840-842, 1978.
- [14] B. Pittel and H. Rubin, How Many Random Questions Are Necessary to Identify n Distinct Objects?, *J. Combinatorial Theory, Ser. A*, 55, 292-312, 1990.
- [15] H. Prodinger, How to Select a Loser, TU Wien, preprint.
- [16] W. Szpankowski, Solution of a Linear Recurrence Equation Arising in the Analysis of Some Algorithms, *SIAM J. Alg. Disc. Methods*, 8, 233-250, 1987.
- [17] W. Szpankowski, Patricia Tries Again Revisited, *JACM*, 37, 691-711, 1990.
- [18] W. Szpankowski and V. Rego, Yet Another Application of a Binomial Recurrence. Order Statistics, *Computing*, 43, 401-410, 1990.