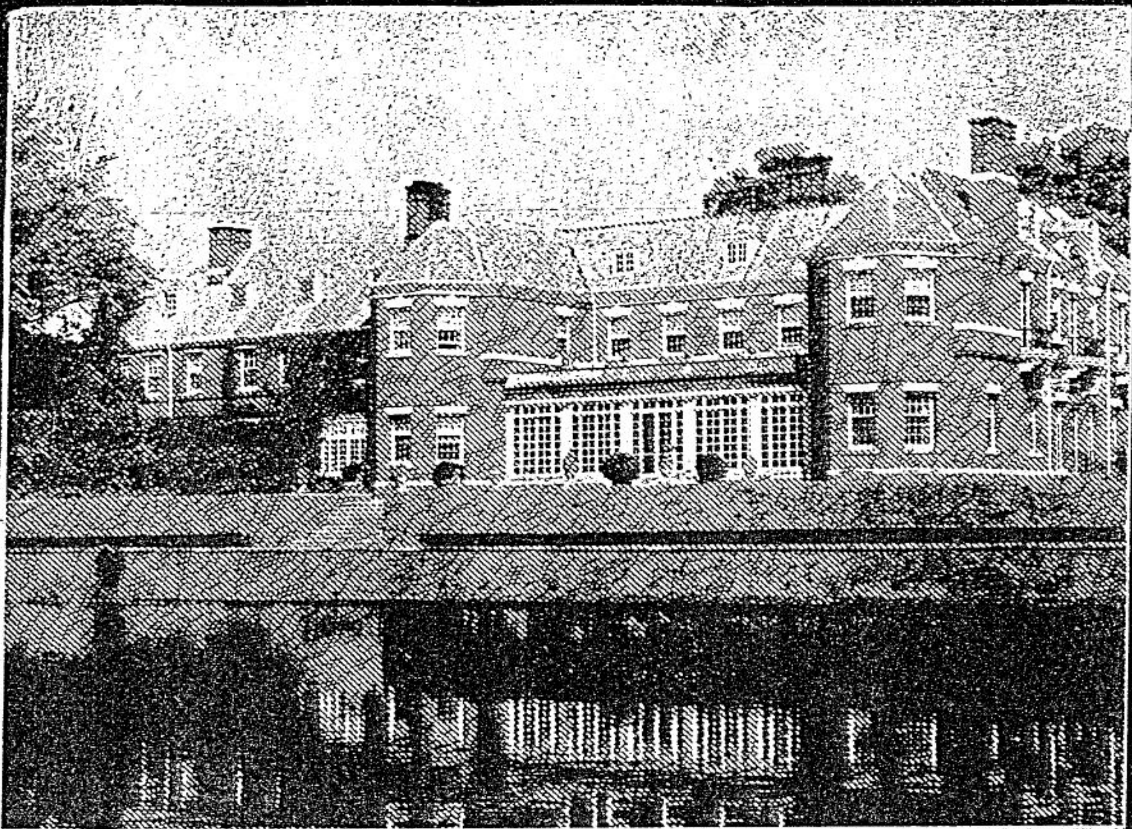

Proceedings

EIGHTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

October 8-10, 1980

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

DYNAMIC DATA STRUCTURES : FINITE FILES, LIMITING PROFILES AND VARIANCE ANALYSIS

L. CHENO[1], P. FLAJOLET[2], J. FRANÇON[3], C. PUECH[4], J. VUILLEMIN[5]

[1]Ecole Normale Supérieure, [2]INRIA, [3]Université de Strasbourg,

[4]Université René Descartes, [5]Université Paris-Sud.

Abstract : Dynamic data structures for priority queues, dictionaries... are analyzed under sequences of arbitrary insert, delete and query operations in various contexts : when the universe of keys is finite both exact and asymptotic estimates are provided ; when the universe of keys is infinite, a new asymptotic formula is given which makes it possible to analyze rather complex data structures including binomial queues. We conclude by further showing how to derive variance estimates for simple structures.

INTRODUCTION

*Dynamic data structures (dds) can be subjected to various operations resul*ting in a possible increase or decrease in size. The oldest such structures are certainly linked lists considered in the mid fifties. Lists have the advantage of requiring hardly more storage (one extra pointer per key) than the quantity of information they are holding at each stage. Trees have the further advantage of allowing a more efficient logarithmic search. The importance and use of these structures for performing dictionary, priority queue, linear list of partition ("union-find") operations are too well-known to be recalled here and are described for instance in [5]. Surprisingly enough, analysis techniques -that constitute the object of the present paper- have taken a much longer time to develop and have received comparatively little attention. The reason can probably be assigned to the difficulty of finding "natural" statistics, and, what is more compelling, to the difficulty of carrying out the analyses themselves due to the inherent complexity of the underlying enumeration problems (it is usually a non trivial task to determine the number of configurations corresponding to a given cost). The common pattern of all the existing dynamic analyses is to consider for each integer n the set of all possible input sequences of length n, or a sample subset taken as representative if the former is infinite. Let Ω_n be the finite set of sequences of length n on which the analysis is performed. For a given data structure S together with its companion algorithms and to each input sequence sometimes also called request sequence of sequence of operations s is associated a certain cost usually measured by the amount of time or storage resource required. Denoting this cost by $\mathrm{cost}^S(s)$, the average cost of a sequence of n inputs (or requests) is naturally defined as :

$$\overline{K}_n^S = \frac{1}{\mathrm{card}\ \Omega_n} \sum_{s\in\Omega_n} \mathrm{cost}^S(s).$$

This quantity will be called the integrated cost of data structure S under sequences of n requests (of type Ω).

The present paper is in the line of previous works by the authors [3] to which the reader is referred for some definitions and a more complete bibliography. We are interested in the following questions not considered in other works :

1) Analyze dictionaries, priority queues... under the set of all possible input sequences when the "universe" of possible keys is finite. This models situations where all transactions are relative to a fixed reference file. The case of priority queues appears for instance in an operating system with a predetermined set of users each enjoying a certain priority.

2) *Analyze more complex data structures than have been previously conside*red : the difficulty then is the rather heavy combinatorial expressions for integrated costs which may soon prove intractable. We prove here the existence of limit distributions for the profiles of histories (sampled input sequences). The approach makes it possible to rederive in a simple way asymptotic expressions for all known integrated costs of priority queues. *We demonstrate its usefulness by providing a complete study of binomial* queues. The analysis reveals rather deep periodicity phenomena in the structure under the effect of arbitrary sequences of inputs.
3) *Study the dispersion of costs of dds under sequences of operations.* We explicit the case of memory occupation for most structures as well as the number of comparisons for sorted and unsorted list implementations of *priority queues and dictionaries.*

About the methods employed , the keystone is the use of continued fractions and orthogonal polynomials whose first appearance in the context of dynamic structures goes to [2], [3]. Profile analysis and its applications branch off with the use of analytical methods, some of them close to works on register allocation or sorting networks. Variance analysis is related to the domain of the so-called "q-analogs" an active domain of combinatorial theory.
In this brief survey, we hope to convince the reader that the analysis of even the simplest algorithms (sorted and unsorted lists) in a dynamic context is a very rich field with a host of interesting combinatorial as well as analytic implications.

PRELIMINARY DEFINITIONS

We consider in the sequel operations of the following three types : Insertions (I) ; Deletions (D) ; Queries (Q).

Let U be the set of all possible keys -the "universe" of keys-. A *sequence of operations of length n relative to U is a sequence of the type* $\omega_1(k_1)\omega_2(k_2)\ \dots\ \omega_n(k_n)$ where $\omega_i \in \{I,D,Q\}$ and $k_i \in U$. Various limitations can be imposed on the admissible sequences of operations, giving rise to dif*ferent data types ; the types we consider here are :*
a) dictionaries (DICT) where operations and keys at any stage are unrestricted ;
b) *dictionaries without queries whose sequences of operations are also* isomorphic to linear lists (LL) : here the ω's can only be I or D ;
c) priority queues (PQ) where operations can be either I or D but a deletion only operates at each stage on the smallest key present in the structure.

To be meaningful a sequence of operations together with all its prefixes must contain more I's than D's (no deletion on an empty file can occur and we shall assume this natural condition to be satisfied). The foregoing analyses will be performed with respect to various sequences of operations each comprising as many I's as D's (thus preventing the analysis to be biased towards insertions). In each case we let Ω_n denote the set of all such sequences of length n, and let H_n denote card Ω_n.
We are interested in the behaviour of data structures w.r.t. to either time or storage, under sequences of operations in Ω_n. To each particular structure is associated the set of its unitary costs CI_k, CD_k, CQ_k where $C\omega_k$ is the cost of operation ω performed on a file of size $k^{\dagger}$.

† In applications, the cost of an operation is not only a function of the size of the file but also of the particular organization of keys resulting *from past operations. It can however be proved that such coefficients* $C\omega_k$ exist for structures that stay "random" under deletions [4]. The $C\omega_k$ are then themselves computable as certain averages over configurations of the structure.

I - FINITE FILES

When the universe of keys is finite, say with cardinality N, the set of allowable input sequences of length n is also finite and we have three different types of statistics PQ/N/, LL/N/, DICT/N/. To estimate integrated costs we first need to count the number of possible request sequences, card $\Omega_n = H_n$. The reader may try to convince himself that

$$H_{2n}^{PQ/2/} = 3^n - 3^{n-1} \quad \textit{and} \quad H_n^{DICT/3/} = \frac{1}{8}6^n + \frac{3}{8}4^n + \frac{3}{8}2^n.$$

Theorem 1^A : [*Generating functions for sequences of operations*]

a) *For* DICT/N/, *i-e dictionaries over a universe of* N *keys*

$$\sum H_n \frac{z^n}{n!} = \left(\frac{e^{2z}+1}{2}\right)^N \quad \text{and} \quad \sum H_n z^n = \sum_j \frac{\binom{N}{j}}{1-2jz} \; ;$$

b) *For* LL/N/, *i-e dictionaries without queries over* N *keys*

$$\sum H_n \frac{z^n}{n!} = \left(\frac{e^{z}+e^{-z}}{2}\right)^N \quad \text{and} \quad \sum H_n z^n = \sum_j \frac{\binom{N}{j}}{1-(N-2j)z} \; ;$$

c) *For* PQ/N/, *i-e priority queues*

$$\sum H_n z^n = \frac{He_N(z)}{He_{N+1}(z)} \quad \textit{where } He_m(z) \textit{ is the } m\text{-th } \textit{Hermite}$$

polynomial : $He_m(z) = \sum_r \frac{m!}{r!(m)2r)!} \left(-\frac{z^2}{2}\right)^r$. □

Let $H_{o,k,n}$ denote the number of extended sequences of operations resulting in a structure of size k i-e comprising k more I's than D's (thus $H_n = H_{o,o,n}$). Then

Theorem 1^B : [*Generating functions for extended sequences of operations*]

a) *For* DICT/N/ : $\sum H_{o,k,n} \frac{z^n}{n!} = \binom{N}{k} e^{Nz} \cosh^{N-k} z \sinh^k z$;

b) *For* LL/N/ : $\sum H_{o,k,n} \frac{z^n}{n!} = \binom{N}{k} \cosh^{N-k} z \sinh^k z$;

c) *For* PQ/N/ : $\sum H_{o,k,n} z^n = \frac{N!}{(N-k)!} \frac{He_{N-k}(z)}{He_{N+1}(z)}$. □

These results rely on the use of the continued fraction theorem of Flajolet [2] ; the three finite fractions relative to the generating functions $\sum H_n z^n$ of theorem 1^A are

DICT/N/ :

$$\cfrac{1}{1-Nz-\cfrac{N.1z^2}{1-Nz-\cfrac{(N-1)2z^2}{\cdots}}} \; ;$$

LL/N/ :

$$\cfrac{1}{1-\cfrac{N.1z^2}{1-\cfrac{(N-1)2z^2}{\cdots}}} \; ;$$

PQ/N/ :

$$\cfrac{1}{1-\cfrac{Nz}{1-\cfrac{(N-1)z}{\cdots}}}$$

The expressions given above yield expressions for the $H_{o,k,n}$, H_n... by combinatorial sums. For example :

$$H_n^{DICT/N/} = 2^{n-N}\left[N^n + \binom{N}{1}(N-1)^n + \binom{N}{2}(N-2)^n + \ldots\right]$$

Given a data structure with unitary costs CI_k, CD_k, CQ_k, we wish to compute the corresponding integrated costs $\overline{K}_n = K_n/H_n$ where $K_n = \sum_{u\in\Omega_n} \text{cost}(u)$. We assume for ease of exposition that $CI_k = CD_k = CQ_k = C_k$.

Theorem 2^A : [*Integrated costs for priority queues over N elements*]
For priority queues, the generating function of total costs $K(z) = \sum K_n z^n$ *is a rational fraction given by*

$$K(z) = \frac{N!}{\left(He_{N+1}(z)\right)^2} \sum_k C_{N-k} \frac{z^{2N-2k}}{(N-k)!} \left(He_k(z)\right)^2 . \qquad \square$$

Similar rational expressions can also be given for linear lists and dictionaries. In these last two cases, it may also prove convenient to operate with exponential generating functions taking

$$\hat{K}(z)' = \sum K_n \frac{z^{n+1}}{(n+1)!} \quad \text{and} \quad C(u) = \sum_k C_k \binom{N}{k} u^k$$

as generating functions of integrated costs and unitary costs.

Theorem 2^B : [*Integral transforms for integrated costs*] : *The generating functions of unitary and integrated costs are related by* :

$$\text{DICT/N/} : \hat{K}(z) = \frac{\left(e^{2z}+1\right)^N}{2} L^{/N/}[C(u)] \; ; \; \text{LL/N/} : \hat{K}(z) = \cosh^N z \; L^{/N/}[C(u)]$$

where the linear transform $L^{/N/}$ *is* :

$$L^{/N/}[C(u)] = 2\int_0^{\tanh^2 z/2} \frac{C(u)}{(1+u)^N(1-u)} \frac{du}{\sqrt{(1+u)^2 \tanh^2 z - 4u}} . \qquad \square$$

As in the other cases discussed in [3], *this last theorem reduces the* determination of integrated costs to the evaluation of integrals of elementary functions relayed by the computation of Taylor expansions. Both theorem 2^A and theorem 2^B permit easy numerical tabulations of the integrated costs of an arbitrary structure.

As an application these results enable us to compare two priority queue *organizations A and B with corresponding integrated costs* $\overline{K}_n^A$ *and* $\overline{K}_n^B$ *under* the PQ/N/ statistics.It can be shown that the quantities $\overline{K}_n^A/n$ which represent the cost of a random operation in a random operation sequence of length n tend, as n gets large, to a constant $\overline{K}_\infty^A$ which, in a way, represents the "steady-state" cost of an operation in A under the statistics PQ/N/.

Theorem 2^C : [*Steady state cost under* PQ/N/]
The steady state cost of an operation of algorithm A *under the statistics* PQ/N/ *is* $\overline{K}_\infty^A = \sum_{0\le k\le N} C_k^A \pi_k$ *where* :

$$\pi_k = \frac{N!}{\alpha \hat{He}_N(\alpha)\hat{He}'_{N+1}(\alpha)} \frac{\alpha^k}{k!} \hat{He}^2_{N-k}(\alpha) \; ;$$

here $\hat{He}(z) = He(\sqrt{z})$ *and* α *the (positive real) root of* $\hat{He}_{N+1}(z)$ *with smallest modulus.* $\square$

Notice that the coefficient π_k is nothing but the limiting probability that an operation takes place at level k in the course of a sequence of length 2n (see in section 2 the related notion of profile).
Theorem 2^C gives a numerical algorithm that suffices in all cases to compare two different data structures.

A treatment of linear lists and dictionaries could be given along very similar lines.

II - PROFILES

We now consider the case where the universe of keys U is infinite. The sampling technique introduced by Françon [4] amounts to selecting one representant in each order-isomorphism class of input sequences. Canonical representants called histories are defined by retaining at each stage only the order of the key operated upon w.r.t. keys already in the structure. For example , for priority queue data type, a history is $I_0 I_0 I_2 D_0 D_0 D_0$ which represents any sequence of operations obeying the pattern : insert a key ; insert another key smaller than the first one (it then has rank 0 w.r.t. the first key), insert another key larger than the two previous ones (it then has rank 2 w.r.t. to these two keys), delete minimum... ; such a history is a canonical represent of say I(2.7) I(0.5) I(3.1) D_{min} D_{min} D_{min}.

We use the same notations as before : H_n represents the number of histories (from 0 to 0) of length n, K_n represents the total cost of all histories of length n and $\overline{K}_n = K_n/H_n$ is the integrated cost of sequences of length n.

It has been shown that priority queue histories lead to the continued fraction

$$\sum_{n\geq 0} H_n z^n = \cfrac{1}{1-\cfrac{1z^2}{1-\cfrac{2z^2}{1-\cfrac{3z^2}{\cdots}}}}$$

and enumerations again involve the Hermite polynomials. From it follows the simple enumeration $H_{2n} = 1.3.5\ldots(2n-1) = n?$.

We recall the following result which has been used to analyze a variety of simple structures :

Theorem 3^A : [*Integrated cost of priority queues*] *The generating function* $\tilde{K}(z) = \sum_{n\geq 0} K_{2n} \frac{z^n}{n!}$ *of the integrated cost of priority queues is related to the generating function of individual costs* $C(u) = \sum_{k\geq 0} (CI_k + CD_{k+1}) u^{k+1}$ *by*

$$\tilde{K}(z) = \frac{1}{\sqrt{1-2z}}\, C\left(\frac{z}{1-z}\right) . \qquad \square$$

Setting $C_k = CI_k + CD_{k+1}$, the above expression is equivalent to combinatorial expressions of integrated costs :

$$K_{2n} = \sum_{m\leq k\leq n} C_k\, 2^{m-n} \binom{n-m-1}{k-1} \frac{\binom{2m}{m}}{\binom{2n}{n}} \quad \text{whence the } \overline{K}_{2n} = K_{2n}/H_{2n}.$$

These expressions are already difficult to handle when $C_k = 1+\frac{1}{2}+\frac{1}{3}\ldots+\frac{1}{k}$. The following result gives a very simple way of computing the first terms in the asymptotic expansion of $\overline{K}_{2n}$:

Theorem 3^B : *For smoothly growing individual costs* C_k, *the integrated costs satisfy*

$$\overline{K}_{2n} = n \int_0^{1/2} C_{\lfloor n\phi\rfloor} \frac{d\phi}{\sqrt{1-2\phi}} \left(1+O\left(\frac{1}{n}\right)\right) . \qquad \square$$

Theorem 3 tells us that weighting corresponding to the passage of individual costs to integrated costs can be described asymptotically by means of an integral form (Previous ways of computing integrated costs required a recourse to generating functions). The proof is achieved by considering the quantities $\pi_{k,n}$ which are the multipliers of C_k in the expression of $\overline{K}_{2n}$; $\pi_{k,n}$ represents the probability of an operation being performed on a file of size k in the course of a random history of length 2n. It is shown that the $\pi_{k,n}$ have a limit distribution with density $\frac{1}{\sqrt{1-2\phi}}$.

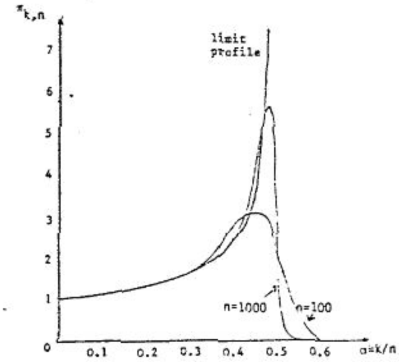

Figure 1 : The limit distribution of $\pi_{k,n}$ and the two cases n=100, n=1000.

The emergence of the limit distribution of the $\pi_{k,n}$ is demonstrated on figure 1 in the cases when n=100 and n=1000. With theorem 3 we readily rederive the values of $\overline{K}_{2n}$ corresponding to individual costs of a simple form :

$$C_k = 1 \Rightarrow \overline{K}_{2n} \sim n \ ;$$

$$C_k = k \Rightarrow \overline{K}_{2n} \sim \frac{n^2}{3} \ ;$$

$$C_k = 1 + \frac{1}{2} + \ldots + \frac{1}{k} \Rightarrow \overline{K}_{2n} \sim n \log n$$

The main interest of theorem 3, besides its simplicity, is to make possible analysis of rather complex structures. We study the case of binomial queues, one of the most efficient administration policies for priority queues, which was discovered by Vuillemin [6].
The structure is based upon a decomposition into blocks of size 1,2,4,8,... .
The analysis of individual costs in a good implementation of binomial queues has been performed by Brown [1] who showed :

Theorem : [*Individual costs for binomial queues*]
Let $k = \sum b_i 2^i$ *be the binary decomposition of* n (b_i = 0 or 1). *Then the undividual costs of binomial queues are* $CI_k = v_2(k)$ *and* $CD_{k+1} = \sigma(k) - v_2(k)$ *where* $v_2(k) = \min\{i | b_i \neq 0\}$ *is the number of trailing zeros in the binary representation of* k *and* $\sigma(k)$ *is an arithmetical function given by*

$$\sigma(k) = \frac{1}{k} \sum_i i\, b_i 2^i. \qquad \square$$

We thus have in this case $C_K = CI_k + CD_{k+1} = \sigma(k)$. Although an expression for $C(u) = \sum c_k u^k$ is available as a Lambert series, it proves rather difficult to work with. Accordingly, $\sigma(k)$ has a rather erratic behaviour reflecting the steep threshold effects for values of k that are powers of 2. The approach through theorem 3 proves extremely useful in this context. We first have :

Theorem 4^A : [*Analytic expression of* $\sigma(k)$] *The function* $\sigma(2p)$ *is expressible as*

$$\sigma(2p) = \log_2 p + G(\log_2 p)$$

where G *is a bounded periodic function with period* 1, *with Fourier coefficients :*

$$g_o = \frac{1}{2} + \frac{\gamma - 1}{\log 2} \; ; \; g_k = \frac{1}{\log 2} \frac{\zeta(1+\chi_k)}{1+\chi_k} \quad (\chi_k = \frac{2ik\pi}{\log 2}) \; . ^{\dagger}$$ □

Function G is highly discontinuous. The above expression can however be combined with theorem 3 to yield.

Theorem 4^B : [*Integrated cost of binomial queues*] *The integrated cost of 2n operations on a binomial queue is* $\overline{K}_{2n} = n \log_2 n + n\, P(\log_2 n) + o(n)$ *where* P *is a continuous function with period* 1, *whose Fourier coefficients* p_k *are*

$$p_0 = \frac{1}{2} + \frac{\gamma - 3}{\log 2} \; ; \; p_k = \frac{1}{\log 2} \frac{\Gamma(1+\chi_k)^2}{\Gamma(2(1+\chi_k))} \frac{\zeta(1+\chi_k)}{1+\chi_k} \; .$$ □

These estimates are in good agreement with the actual integrated costs. On figure 1 below we have plotted the values of

$Q(n) = \frac{\overline{K}_{2n}}{n} - \log_2 n$, and this last term rapidly conforms to $P(\log_2 n)$.

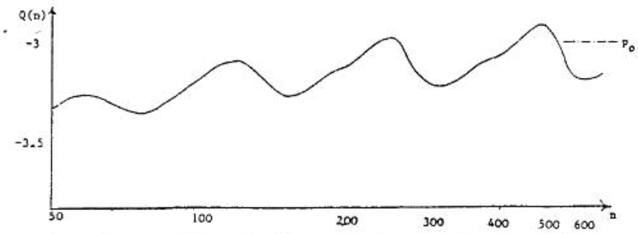

Figure 2 : The quasi-periodic term in the integrated cost of binomial queues.

Similar treatments could be given for dictionaries and linear lists.

III - VARIANCES

We have shown in the previous two sections ways of estimating costs either exactly or asymptotically in various contexts of requests. We here show that the continued fraction approach can be used to also attain results about typicality of the average case. More precisely, we consider five problems hereafter called MemPQ, MemDICT, TimePQ-SL, TimePQ-UL, TimeDICT-SL. Here Mem stands for memory, PQ and DICT retain their meaning as priority queue and dictionary types, SL is the sorted list and UL the unsorted list structure. So, for instance TimePQ-SL refers to the time cost of the sorted list implementation of priority queues, MemPQ refers to the total storage occupation measured in the number of keys stored under the statistics of priority queues histories... . These various problems are studied in the context of an infinite set of keys as in section 2. We shall present the line of attack on TimePQ-SL, i-e number of comparison (*comp*) in the sorted list implementation of priority queues under sequences of operations.
Given a priority queue history, the number of resulting comparisons in the sorted list implementation is readily determined : when an insertion operates with rank r, the number of comparisons performed to place the element in the structure is plainly r+1 ; a deletion requires no comparison

† Γ and ζ refer resp. to the Euler gamma function and the Riemann zeta function ; γ is the Euler constant.

at all since the minimum element always occurs first in the structure. The number of comparisons associated with $h=I_0 I_0 I_2 I_1 I_4 D_{min} D_{min} I_2 D_{min} D_{min} D_{min} D_{min}$

is $comp(h) = (0+1) + (0+1) + (2+1) + (1+1) + (4+1) + (2+1) = 13$.

Let $H_{n,c}$ be the number of priority queue histories of length n leading to c comparisons in SL. We are interested in the distribution of *comp*, and more specifically to its mean and variance.

From results recalled in section 2, we know that $\sum_c H_{2n,c} = 1.3.5\ldots(2n-1) = n?$, representing the total number of priority queue histories.

The $H_{2n,c}$ have a number of interesting combinatorial properties :

$$H_{2n,c} \neq 0 \quad \text{iff } n \leq c \leq \frac{n(n+1)}{2} \quad ; \; H_{2n,n} = \frac{1}{n+1}\binom{2n}{n} \quad ; \quad H_{2n,\frac{n(n+1)}{2}} = 1.$$

In the sequel we shall work with the polynomials $H_{2n}(q) = q^{-n}\sum_c H_{2n,c}\, q^c$.

The parameter *comp* belongs to the category of cumulative parameters in the analysis of algorithms. Such parameters have quadratic worst case and occur for instance in the analysis of insertion sort, bubble sort, tree sort, quisksort... . From an analytical stand point the difficulty is that a series like $\sum_n H_{2n}(q)\, z^n$ of $\sum_n H_{2n}(q)\, \frac{z^n}{n!} \ldots$,

can in no way be obtained as a combination of elementary functions since the degree of $H_{2n}(q)$ is $\frac{n(n-1)}{2}$.

Appealing again to the continued fraction theorem, one can prove that :

$$H(z;q) = \sum H_{2n}(q) z^{2n} = \cfrac{1}{1-\cfrac{[1]z^2}{1-\cfrac{[2]z^2}{1-\cfrac{[3]z^2}{\cdots}}}} \qquad \text{where } [s] = 1+q+q^2+\ldots+q^{s-1} .$$

This continued fraction reduces to the continued fraction of section 2 when q=1 of which it is said to be a q-analog.

The mean and variance of *comp* over histories of length 2n are given by :

$$\mu_{2n} = \frac{1}{n?}\left[\frac{dH_{2n}}{dq}(q)\right]_{q=1} + n \quad ; \; \sigma^2_{2n} = \frac{1}{n?}\left[\frac{d^2}{dq^2} H_{2n}(q)\right]_{q=1} - \mu^2_{2n} + \mu_{2n},$$

but it is to be noted that H(z;q) together with its derivates at q=1 strongly diverge.

The convergents of H(z;q) are finite fractions which can be expressed as $\frac{P_r(z;q)}{Q_r(z;q)}$ with P_r and Q_r polynomials in z and q. The denominator polynomials Q_r satisfy a classical recurrence relation ; they are related to q-analogs of the Hermite polynomials.

A Eulerian generating function of the reciprocal polynomials of Q_r *defined by :* $K(t,z;q) = \sum_{r\geq 0} Q_{r-1}\left(\frac{1}{z}|q\right) \frac{z^r t^r}{[1][2]\ldots[r]}$,(this reduces to $e^{-(t^2/2)+tz}$ when q=1) satisfies the difference equation $\frac{K(qt,z;q)-K(t,z;q)}{(q-1)t} = (z-t)K(t,z;q)$.

This equation reduces to a classical differential equation when q=1 ; it can be "solved" expressing K either as an infinite product, or in terms of the q-exponential. It suffices here to notice that this functional equation makes it possible to determine the values of

$$\left[\frac{d}{dq}\ Q_r(z;q)\right]_{q=1} \ , \ \left[\frac{d^2}{dq^2}\ Q_r(z;q)\right]_{q=1} \ \ldots$$

after a rather difficult computation. Using orthogonality relations that hold between the Q's and the H's further allows determination of

$$\left[\frac{d}{dq}\ H_{2n}(q)\right]_{q=1} \ , \ \left[\frac{d^2}{dq^2}\ H_{2n}(q)\right]_{q=1} \ \ldots$$

and we have the very simple result :

$$a)\ \mu_{2n} = \frac{n(n+5)}{6} \quad ; \quad b)\ \sigma^2_{2n} = \frac{n(n-1)(n+3)}{45} .$$

These results are related to combinatorial results by Touchard and Riordan relative to chord intersection problems. We have here a unified approach which leads to a solution of all the problems mentioned above. Also moments of higher order can be determined. We state :

Theorem 5 : *The mean and variance of memory for dictionary and priority queue histories are given by :*

$$\text{MemPQ :} \quad \mu_{2n} = \frac{n(2n+1)}{3} \qquad \sigma^2_{2n} = \frac{n(n-1)(2n+1)}{45}$$

$$\text{MemDICT :} \quad \mu_n = \frac{(n-1)(n+1)}{6} \qquad \sigma^2_n = \frac{20n^4-278n^3+1472n^2-3073n+2047}{180} .$$

The mean and variance of time for the sorted list implementation of dictionaries and priority queues are given by

$$\text{TimePQ-SL :} \quad \mu_{2n} = \frac{n(n+5)}{6} \qquad \sigma^2_{2n} = \frac{n(n-1)(n+3)}{45}$$

$$\text{TimeDICT-SL :} \quad \mu_n = \frac{(n-1)(n-2)}{12} \qquad \sigma^2_n = \frac{(n-2)(2n^2+11n-1)}{360} . \qquad \square$$

For instance the proof of MemPQ uses another q-analog of Hermite polynomials whose associated generating function satisfies the difference-differential equation

$$\frac{\partial K(t,z;q)}{\partial t} = zK(t,z;q) - qK(qt,z;q).$$

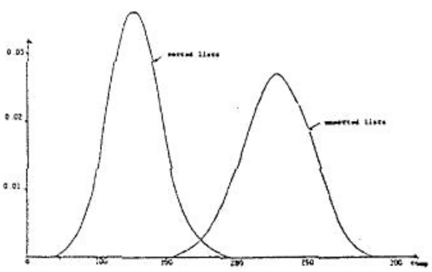

Figure 3 : Distribution of costs for sorted vs. unsorted implementations of priority queues.

The problem TimePQ.UL of the time complexity for unsorted *lists actually reduces to* MemPQ. Figure 3 gives the comparative distributions in number of comparisons of sorted versus unsorted lists under the PQ statistics when n=25.

REFERENCES

[1] M.R. Brown,
Implementation and analysis of binomial queues algorithms,
SIAM J. Comput., vol. 7, n° 3 (1978), pp. 298-319.

[2] P. Flajolet,
Combinatorial aspects of continued fractions,
Discrete Maths, (1980), to appear.

[3] P. Flajolet, J. Françon, J. Vuillemin,
Sequence of Operations Analysis of Dynamic Data Structures,
Journal of Algorithms, 1, (1980), pp. 111-141.

[4] J. Françon,
Combinatoire des Structures de Données
Thèse, Université Louis Pasteur, Strasbourg (1979).

[5] D. Knuth,
The Art of Computer Programming,
Vol. 3, (1973).

[6] J. Vuillemin,
A Data Structure for Manipulating Priority Queues,
Comm. ACM 21, n° 4, pp. 309-315.

Mailing address : Ph. FLAJOLET, INRIA, 78150 ROCQUENCOURT (FRANCE).