Maximum Queue Size and Hashing with Lazy Deletion

Claire M. Kenyon[1] and Jeffrey Scott Vitter[2]

Abstract. We answer questions about the distribution of the maximum size of queues and data structures as a function of time. The concept of "maximum" occurs in many issues of resource allocation. We consider several models of growth, including general birth-and-death processes, the $M/G/\infty$ model, and a non-Markovian process (data structure) for processing plane-sweep information in computational geometry, called "hashing with lazy deletion" (HwLD). It has been shown that HwLD is optimal in terms of expected time and dynamic space; our results show that it is also optimal in terms of expected *preallocated space*, up to a constant factor.

We take two independent and complementary approaches: first, in Section 2, we use a variety of algebraic and analytical techniques to derive exact formulas for the distribution of the maximum queue size in stationary birth-and-death processes and in a nonstationary model related to file histories. The formulas allow numerical evaluation and some asymptotics. In our second approach, in Section 3, we consider the $M/G/\infty$ model (which includes $M/M/\infty$ as a special case) and use techniques from the analysis of algorithms to get optimal big-oh bounds on the expected maximum queue size and on the expected maximum amount of storage used by HwLD in excess of the optimal amount. The techniques appear extendible to other models, such as $M/M/1$.

1. Introduction. Queueing phenomena are widespread in the fields of operating systems, distributed systems, and performance evaluation. Queues are also a natural way to model the size of classical dynamic data structures, such as buffers, dictionaries, sets, stacks, queues, priority queues, and sweepline structures. As a consequence, many statistical properties of queues have been investigated, such as their expected size and variance. Yet, very little was known about the *maximum size* of queues over a given period of time. If the size of the queue represents the amount of resource used by a computer program or a systems component, then such information is important for making intelligent decisions about preallocating resources.

Another motivation for our study was the need to develop and analyze practical space-efficient methods for processing sweepline information. Some work in this area has been done by Van Wyk and Vitter [1986], Morrison et al. [1987], and Ottmann and Wood [1986], but as the latter point out, "Surprisingly there has been little theoretical investigation of space-economical plane-sweep algorithms even though such algorithms have significant practical applications." Ottmann and Wood [1986] do not investigate the maximum number of items cut by the sweepline; they express the running times of their algorithms in terms of the maximum number. Our approach in this paper is to examine the distribution of the maximum number of cut items, based on several popular input models, and in addition show that the "hashing with lazy deletion" (HwLD) algorithm introduced by Van Wyk and Vitter [1986] is extremely practical and optimum in both average running time and preallocated space.

We develop new methods and obtain several results about the distribution of the maximum queue size, under several models of growth. We study stationary birth-and-death processes, and are particularly interested in $M/M/\infty$ and the more general $M/G/\infty$ queues, which model the amount of plane-sweep information as a function of time. We also concentrate on HwLD, which is a non-Markovian queueing model corresponding to the space usage of the algorithm by the same name. In addition we study a nonstationary model corresponding to histories of priority queues.

Plane-sweep algorithms process a sequence of items over time; at time t the data structure stores the items that are "living" at time t. Let us think of the ith item as being an interval $[s_i, t_i]$ in the unit interval, containing a unique key k_i of supplementary information. The ith item is "born" at time s_i, "dies" at time t_i, and is "living" at time t when $t \in [s_i, t_i]$. The data structure must be able to support the dynamic operation of searching the living items based on key value. It is natural to think of the data structure as a queue, as far as size is concerned. Let us denote the queue size at time t by $Need(t)$, the number of items that need to be included in the data structure. If we think of the items as horizontal intervals, then $Need(t)$ is just the number of intervals "cut" by the vertical sweepline at position t. In a typical application, we may have 10^6 intervals in the time range $[0, 1]$, with $E(Need) = 10^3$; that is, only the square root of the total number of items tends to be present at any given time [Szymanski and Van Wyk, 1983]. It is thus very inefficient to devote a separate storage location to every item; the data structure should be dynamic.

In HwLD, items are stored in a hash table of H buckets, based upon the hash value of the key. The distinguishing feature of HwLD is that an item is not deleted as soon as it dies; the "lazy deletion" strategy deletes a dead item only when a later insertion accesses the same bucket. The number H of buckets is chosen so that the expected number of items per bucket is small. HwLD is thus more time-efficient than doing "vigilant-deletion," at a cost of storing some dead items.

Let $Use(t)$ be the number of items in the HwLD data structure at time t. It is shown by Van Wyk and Vitter [1986] for the $M/M/\infty$ model that at any given time t we have

$$E(Use) = E(Need) + H = \frac{\lambda}{\mu} + H,$$

where λ is the birth rate of the intervals and $1/\mu$ is the average lifetime per item. The amount of wasted space is equal to the number H of buckets. A possible choice of H is $H = \Theta(E(Need))$, so that the expected amounts of space and time used by HwLD are optimal, up to a constant factor. (In practice, the computer memory space used by HwLD if often *less than* the space used by "vigilant-deletion" strategies, because the latter are typically based on balanced trees and priority queues, which require more storage overhead (pointer information) per item.) It was conjectured by Van Wyk and Vitter [1986] that

$$E\left(\max_{0 \le t \le 1} \{Need(t)\} \right) = O\left(\frac{\lambda}{\mu}\right),$$

$$E\left(\max_{0 \le t \le 1} \{Use(t)\} - \max_{0 \le t \le 1} \{Need(t)\} \right) = O(H),$$

which would prove that HwLD is also optimal in terms of *preallocated* storage. A system of equations for the distribution of $\max_t\{Need(t)\}$ and for the degenerate $H = 1$ distribution of $\max_t\{Use(t)\}$ in equilibrium for the M/M/∞ model was recently developed by Morrison et al. [1987]. This system can be used to get numerical data. Both distributions are nearly identical, because when $H = 1$ we have $\max_{t > t^*}\{Need(t)\} = \max_{t > t^*}\{Use(t)\}$, where t^* is the birthtime of the first item to enter the queue after time $t = 0$.

   In this paper we attain an array of results about the maximum queue size using two independent approaches. In the first approach, described in the next section, we develop several formulas for the distribution of $\max_t\{Need(t)\}$ for general birth-and-death processes (which includes the M/M/∞ process) and for the distribution of $\max_t\{Use(t)\}$ in the general $H > 1$ case of HwLD. We also handle a nonstationary model described by Van Wyk and Vitter [1986]. The formulas provide exact numerical data on the distributions, and in some cases lead to asymptotics as the time interval grows. There is a common underlying structure in the formulas for the different models: the transform of interest in each case is the ratio of consecutive classical orthogonal polynomials.

   In our second approach, described in Section 3, we prove the above conjectures for the general M/G/∞ model, which includes M/M/∞ as a special case. We obtain optimal big-oh bounds on the expected maximum queue size by using non-queueing-theory techniques. We approximate the continuous-time processes $\max_t\{Need(t)\}$ and $\max_t\{Use(t)\}$ by sums of discrete quantities related to hashing, specifically, maximum slot occupancies. (The hashing in our approximation scheme has nothing to do with the hashing inherent in HwLD.) Our techniques also seem applicable to other queueing models, such as M/M/1.

2. Exact Formulas for Maximum Queue Size.   It is convenient to extend the range of time to $[0, T]$ for arbitrary T; the results can be translated back to $T = 1$ later. In the following sections we derive exact formulas for the distribution of the maximum queue size in several models. Our formulas are amenable to numerical calculation and yield asymptotic expressions in some cases.

The problem has been studied previously by Morrison *et al.* [1987] for the special cases of M/M/∞ and the $H = 1$ case of HwLD. However, analysis for the case $H = 1$ cannot be used to get a good bound for when $H > 1$; a corollary of our analysis in Section 3 is that $H \cdot \max_t\{Use_1(t)\}$ is typically greater than $\max_t\{Use(t)\}$ by more than a constant factor, where $Use_1(t)$ is the occupancy of bucket 1 at time t.

A birth-and-death process is a Markov process in which transitions from level k are allowed only to levels $k + 1$ and $k - 1$. We restrict ourselves to continuous time in this exposition. Borrowing notation from HwLD, we define $Need(t)$ to be the level of the process at time t. The infinitesimal birth and death rates at level k are denoted λ_k and μ_k:

$$\Pr\{Need(t + \Delta t) = j \mid Need(t) = k\} = \begin{cases} \lambda_k \Delta t + o(\Delta t) & \text{if} \quad j = k + 1, \\ 1 - (\lambda_k + \mu_k)\Delta t + o(\Delta t) & \text{if} \quad j = k, \\ \mu_k \Delta t + o(\Delta t) & \text{if} \quad j = k - 1, \\ o(\Delta t) & \text{otherwise.} \end{cases}$$

For the special case of the M/M/∞ model we write $\lambda_0 = \lambda_1 = \cdots = \lambda$ and $\mu_k = k\mu$. For the M/M/1 model we write $\lambda_0 = \lambda_1 = \cdots = \lambda$ and $\mu_1 = \mu_2 = \cdots = \mu$. In both cases the arrival process is Poisson, and for the M/M/∞ case the lifespans are exponentially distributed. The reader can consult [Kleinrock, 1975] for further background.

In Sections 2.1–2.5 we derive exact formulas for $\Pr\{\max_{0 \le t \le T}\{Need(t)\} = k\}$ using a variety of algebraic and analytical techniques. The first three sections handle the case of general homogeneous and stationary birth-and-death processes in equilibrium at $t = 0$, the fourth discusses HwLD under the M/M/∞ model in equilibrium at $t = 0$, and the last deals with a nonstationary model.

2.1. Applications of Stack Histories. A Dyck path is a walk in $\mathbf{Z}^2$ above the x-axis such that each step is of the type $(a, b) \to (a + 1, b \pm 1)$. Its level is the maximal y-coordinate reached. Dyck paths are a special case of file histories: they correspond to histories of stacks [Flajolet *et al.*, 1980]. (File histories are discussed further in Section 2.3.) Let ω be a Dyck path going from level i to level l in n steps, and with height constrained to be $\le k$. For each such ω we define $p_\omega(T)$ to be the probability that in time interval $[0, T]$ the successive different states of the process $Need(t)$ correspond exactly to ω, given that $Need(0) = i$.

LEMMA 2.1. *We have*

$$\Pr\left\{ \max_{0 \le t \le T} \{Need(t)\} \le k \right\} = \sum_{i, l, n, \omega} (\Pr\{Need(0) = i\} \cdot p_\omega(T)).$$

As an example of our method, let us consider the M/M/1 model with parameters λ and μ. The equilibrium probabilities are given by $\Pr\{Need = i\} = (\lambda/\mu)^i(1 - \lambda/\mu)$. It remains to calculate the $p_\omega(T)$ terms, which can be expressed as a multiple

integral. In fact, $p_\omega(T)$ does not depend upon the actual shape of ω, but only upon the number of times the path hits the x-axis. Using that gives us $p_\omega(T)$ in simple summation form. Lemma 2.1 can thus be applied to yield an exact expression for $\Pr\{\max_{0 \leq t \leq T}\{Need(t)\} \leq k\}$.

2.2. Applications of Orthogonal Polynomials. We can extend the approach used by Morrison *et al.* [1987] for the M/M/∞ model to general birth-and-death processes. We have

$$\Pr\left\{ \max_{0 \leq t \leq T} \{Need(t)\} \leq k \right\} = \sum_{0 \leq j \leq k} \Pr\{Need(0) = j\} \cdot \int_T^\infty s_{j,k}(t)\, dt,$$

where $s_{j,k}(t)$ is the density of the first-passage time to level k starting from level j.

These densities $\{s_{j,k}\}_{j<k}$ are solutions of a system of integral equations; taking Laplace transforms $\sigma_{j,k}(s)$ we get the following system:

$$\sigma_{j,k}(s) = \sigma_{j,j+1}(s)\sigma_{j+1,j+2}(s) \cdots \sigma_{k-1,k}(s),$$

$$\sigma_{j,j+1}(s) = \frac{\lambda_j + \mu_j\sigma_{j-1,j+1}(s)}{s + \lambda_j + \mu_j}.$$

Let us define $\omega_{j+1}(s)$ to be $1/(\sigma_{0,1}\sigma_{1,2} \cdots \sigma_{j,j+1})$, where $\omega_0(s) = 1$. We find that $\omega_{j+1}(s)$ is a polynomial of degree $j + 1$, and it can be computed by iteration. Hence, $\sigma_{j,k}(s) = \omega_j(s)/\omega_k(s)$ is a rational fraction; its poles are roots of ω_k, and yield $s_{j,k}(t)$ and thus $\Pr\{\max_{0 \leq t \leq T}\{Need(t)\} \leq k\}$. Moreover, when $Need(t)$ is a birth-and-death process, computing the roots of ω_k is an easier task because $\{\omega_j\}$ is a family of orthogonal polynomials, and when T goes to infinity, $\Pr\{\max_{0 \leq t \leq T}\{Need(t)\} \leq k\} \sim Ke^{\alpha T}$, with K a constant and α a root of ω_k with maximal modulus.

Karlin and McGregor [1958] introduce the family of polynomials $\{Q_n(x)\}$ with the properties that $Q_0(x) \equiv 1$ and $-xQ = AQ$, where A is the infinitesimal generator matrix defined by

$$A_{k,j} = \begin{cases} \lambda_k & \text{if } j = k + 1, \\ -\lambda_k - \mu_k & \text{if } j = k, \\ \mu_k & \text{if } j = k - 1, \\ 0 & \text{otherwise.} \end{cases}$$

It turns out that $Q_n(x) = \omega_n(-x)$. This expression gives an extremely simple tool for linking birth-and-death processes to classical families of orthogonal polynomials:

THEOREM 2.1. *For the M/M/1 process we have*

$$(\sqrt{a})^n Q_n(\mu x) = T_n(z) - \frac{1}{\sqrt{a}} T_{n-1}(z),$$

where $a = \lambda/\mu$, $z = -(x - a - 1)/\sqrt{a}$ and $\{T_j(u)\}$ is the family of Chebyshev orthogonal polynomials. For the $M/M/\infty$ process we have

$$Q_j(x) = (-1)^j a^j C_j^{(a)}\left(\frac{x}{\mu}\right),$$

where $\{C_j^{(a)}(u)\}$ is the family of Poisson–Charlier orthogonal polynomials. For several types of linear birth-and-death processes, of the form $\lambda_k = \alpha k + \beta$, $\mu_k = \gamma k + \delta$, $Q_j(x)$ can be expressed in terms of either Laguerre polynomials or Meixner polynomials of the second kind.

General birth-and-death processes can also be related to orthogonal polynomials, using the framework of file histories discussed by Flajolet et al. [1980].

2.3. Applications of Continued Fractions. File histories model the evolution of several classical types of dynamic data structures: stacks (S), priority queues (PQ), linear lists (LL), symbol tables (ST), and dictionaries (D). The data structures are treated as combinatorial objects; their performance characteristics are determined by the *relative* order of the elements they contain, not by the actual values of the elements. Thus, we say that there are $k + 1$ ways of inserting a new element into a dictionary of size k, since there are $k + 1$ "gaps" where the new element can fit in, relative to the k elements already present. The evolution of the data structure is represented as a path in $\mathbf{Z}^2$ (the x-coordinate counts the number of operations, whether they be insertions, deletions, or queries, and the y-coordinate counts the size), where each step is of the type $(a, b) \to (a + 1, b \pm 1)$ (insertion or deletion) or $(a, b) \to (a + 1, b)$ (positive or negative query). To each step we associate a certain choice among the possibilities, each equally likely. For example, in priority queue files, deletions can be performed only for the minimum element, so the number of possibilities for a deletion is 1. Table 2.1 summarizes the number of possibilities for each type of data structure and operation.

Flajolet [1981] showed that the ordinary generating functions of file histories are equal to continued fractions, and to the convergents when the height is constrained. This gives yet another way of expressing $\sigma_{j,k}(t)$, namely, in terms of generating functions of histories. (A related result is Karlin and McGregor's formula for expressing the transition probability $P_{0,0}(t)$ for birth–death processes as a continued fraction [Karlin and McGregor, 1958].)

Table 2.1

	D	PQ	LL	ST	S
Insertions	$k + 1$	$k + 1$	$k + 1$	$k + 1$	1
Deletions	k	1	k	1	1
Positive queries	k	0	0	k	0
Negative queries	$k + 1$	0	0	0	0

For purposes of brevity, let us restrict ourselves to the $M/M/\infty$ model in which $\lambda = \mu$. This process is related to histories of symbol tables, in which the number of possibilities for insertion, deletion, and query at level k are equal to $k + 1$, 1, and k, respectively [Flajolet *et al.*, 1980]. We let $H_{j,k}(t)$ be the ordinary generating function of the number of symbol table histories going from level j to k, and we define $H_{j,k}^{\leq h}(t)$ similarly except with the histories constrained to have height $\leq h$.

Let us consider the bounded process $\lambda_0 = \lambda_1 = \cdots = \lambda_{h-1} = \lambda$, $\lambda_h = 0$, $\mu_k = k\mu$, whose height can never exceed level h (this process can be denoted $M/M/\infty/h$). We define $s_{j,k}^{\leq h}(t)$ to be the associated density function for the first passage time to level k. If we call $\sigma_{i,i-1}^{\leq h}(s)$ the Laplace transform of $s_{i,i-1}^{\leq h}(t)$, then $\sigma_{i,i-1}^{\leq h}(s)$ is the solution of the system

$$\sigma_{i,i-1}^{\leq h}(s) = \frac{i\mu}{\lambda + i\mu + s - \lambda\sigma_{i+1,i}^{\leq h}(s)}$$

for $i < h$. Hence we have

$$\sigma_{i,i-1}^{\leq h}\left(-\mu\left(1 + \frac{1}{s}\right)\right) = i\frac{\mu}{\mu + s} \cdot \cfrac{1}{1 - is - \cfrac{(i+1)s^2}{1 - (i+1)s - \cfrac{(i+2)s^2}{\cdots \cfrac{}{1 - (h-1)s - \cfrac{hs^2}{1 - (h-1)s}}}}},$$

and as $\sigma_{k,l}(s) = \sigma_{k,k-1}(s)\cdots\sigma_{l+1,l}(s)$, if $k > l$, we see that

$$\sigma_{k,l}(s) = (-1)^{k-l-1}(l+1)\frac{\mu}{\mu + s} H_{l+1,k}^{\geq l+1}\left(-\frac{\mu}{\mu + s}\right).$$

As for the upper first-passage time densities, in a similar way we can show that

$$\sigma_{i,i+1}(s) = \frac{\mu}{\mu + s} \cdot \cfrac{1}{1 - iy - \cfrac{iy^2}{1 - (i-1)y - \cfrac{(i-1)y^2}{\cdots \cfrac{}{1 - y - y^2}}}},$$

where $y = -\mu/(\mu + s)$. This gives us

$$\sigma_{i,i+1}(s) = \frac{\mu}{\mu + s} H_{i,i}^{\leq i}\left(-\frac{\mu}{\mu + s}\right),$$

and for $k < l$ this yields

$$\sigma_{k,l}(s) = \sigma_{k,k+1}(s)\sigma_{k+1,k+2}(s) \cdots \sigma_{l-1,l}(s) = (-1)^{l-k-1} \frac{\mu}{\mu+s} H_{l-1,k}^{\leq l-1}\left(-\frac{\mu}{\mu+s}\right).$$

Thus the Taylor coefficients of $\sigma_{k,l}(\mu(-s-1))$ can be interpreted in terms of the number of histories going from level $l-1$ to level k with bounded height.

All file histories seen so far have their height bounded above or below by some constant. This is due to our concentrating on times of first passage through a state l, which implies that level l must be a barrier for the histories: they must not be allowed to go through state l. But if we now remove the constraint of first passage and consider $P_{k,l}(t) = \Pr\{Need(t) = l \mid Need(0) = k\}$, in the same way we now get (with $\pi_{k,l}(s)$ the Laplace transform of $P_{k,l}(t)$)

$$\pi_{k,l}(s) = \begin{cases} (-1)^{l-k}\dfrac{1}{\mu+s}H_{k,l}\left(-\dfrac{\mu}{\mu+s}\right) & \text{if } k \leq l, \\[3mm] (-1)^{l-k}\dfrac{1}{(l+1)(\mu+s)}H_{k,l}\left(-\dfrac{\mu}{\mu+s}\right) & \text{if } k > l. \end{cases}$$

Taking the inverse Laplace transform will finally yield $s_{j,k}(t)$ and $P_{j,k}(t)$.

2.4. Hashing with Lazy Deletion.

The case $H = 1$ in which there is no hashing and a vigilant-deletion strategy is used was analyzed by Morrison et al. [1987]. We can generalize their method to $H = 2$ by considering the appropriate conditional probabilities. For bucket i, we define $Use_i(t)$ and $Need_i(t)$ in the obvious way and define $Waste_i(t) = Use_i(t) - Need_i(t)$. We have $Use(t) = Need_1(t) + Need_2(t) + Waste_1(t) + Waste_2(t)$.

At equilibrium we have, because of the independence between the two buckets,

$$\begin{aligned} \Pr\{N_1(0) &= n_1, W_1(0) = w_1, N_2(0) = n_2, W_2(0) = w_2\} \\ &= \Pr\{N_1(0) = n_1, W_1(0) = w_1\} \times \Pr\{N_2(0) = n_2, W_2(0) = w_2\} \\ &= \frac{(n_1 + w_1)(\lambda/2\mu)^{n_1+w_1}\Gamma(\lambda/2\mu + n_1)e^{-\lambda/2\mu}}{n_1!\,\Gamma(\lambda/2\mu + n_1 + w_1 + 1)} \\ &\quad \times \frac{(n_2 + w_2)(\lambda/2\mu)^{n_2+w_2}\Gamma(\lambda/2\mu + n_2)e^{-\lambda/2\mu}}{n_2!\,\Gamma(\lambda/2\mu + n_2 + w_2 + 1)}. \end{aligned}$$

Let us define

$$\begin{aligned} P_{n_1,w_1,n_2,w_2,h}(T) = \Pr\Big\{ \max_{0 \leq t \leq T} \{Use(t)\} < h \mid N_1(0) = n_1, W_1(0) \\ = w_1, N_2(0) = n_2, W_2(0) = w_2 \Big\}. \end{aligned}$$

Then we have

$$\Pr\left\{\max_{0\le t\le T}\{Use(t)\}<h\right\}$$

$$=\sum \Pr\{N_1(0)=n_1,\ W_1(0)=w_1,\ N_2(0)=n_2,\ W_2(0)=w_2\}P_{n_1,w_1,n_2,w_2,h}(T),$$

where the sum is over the domain $n_1+n_2+w_1+w_2<h$. These probabilities can be computed:

$$P_{n_1,w_1,n_2,w_2,h}(T)=\int_T^\infty s_{n_1,w_1,n_2,w_2,h}(t)\,dt,$$

where $s_{n_1,w_1,n_2,w_2,h}(t)$ is the first-passage time density from the given initial state (n_1,w_1,n_2,w_2) to a state where $Use(t)=h$. It satisfies the following equality:

$$s_{n_1,w_1,n_2,w_2,h}(t)$$

$$=\int_0^t e^{-(\lambda+(n_1+n_2)\mu)u}\left(\frac{\lambda}{2}s_{n_1+1,0,n_2,w_2,h}(t-u)+\frac{\lambda}{2}s_{n_1,w_1,n_2+1,0,h}(t-u)\right.$$

$$+n_1\mu s_{n_1-1,w_1+1,n_2,w_2,h}(t-u)$$

$$\left.+n_2\mu s_{n_1,w_1,n_2-1,w_2+1,h}(t-u)\right)du$$

$$+e^{-(\lambda+(n_1+n_2)\mu)t},$$

if n_1,n_2,w_1, and w_2 are nonnegative numbers whose sum is smaller than h, and 0 otherwise. Taking the Laplace transform we get a closed system of linear equations and, solving, we get the density $s_{n_1,w_1,n_2,w_2}(t)$. This yields a method for obtaining numerically the values of $\Pr\{\max_{0\le t\le T}\{Use(t)\}<h\}$ and hence the distribution and mean of $\max_{0\le t\le T}\{Use(t)\}$. The same tools can be used for getting numerical results in any case where H is a fixed constant.

2.5. Nonstationary Model and Hermite Polynomials. We consider the nonstationary model introduced by Van Wyk and Vitter [1986], in which the $2n$ birthtimes and deathtimes of the n items are independent uniform random variables from the unit interval. The ith item is born at time $\min\{s_i,t_i\}$ and dies at time $\max\{s_i,t_i\}$. The average queue size $E(Need(t))=2nt(1-t)$ attains its maximum $n/2$ at $t=\frac{1}{2}$. The question of interest is to determine the distribution of the random variable $\max_{0\le t\le 1}\{Need(t)\}$. We shall see that it is the same as the height of a priority queue file history as discussed by Flajolet *et al.* [1980].

For a given choice of $(s_1,\ldots,s_n,t_1,\ldots,t_n)$, since the $2n$ values are distinct with probability 1, they can almost surely be sorted into increasing order $u_1<u_2<\cdots<u_{2n}$. We now define an involution $\sigma\in S_{2n}$ by the condition that $\sigma(i)=j$ if

and only if there is a k such that $\{u_i, u_j\} = \{s_k, t_k\}$. An involution σ has no fixed point; for each k, it associates the two values that delimit the life of customer k. Notice that knowledge of σ gives enough information to obtain the value of $\max_{0 \leq t \leq 1}\{Need(t)\}$. The ith transaction on the structure is an insertion if $\sigma(i) > i$, and then the size is increased by one, and it is a deletion if $\sigma(i) < i$, in which case the size is decreased by one.

Since the random variables $s_1, \ldots, s_n, t_1, \ldots, t_n$ are independent and uniformly distributed in the unit interval, all involutions σ with no fixed point are equally likely. But we know that there is a one-to-one correspondence between such involutions and file histories weighted by $q_i = 0$, $a_{i-1}s_i = i$, which are nothing but priority queue file histories. (Here q_i, a_i, and s_i are the number of possibilities for query, insertion, and deletion.) Hence, the problem is equivalent to knowing the distribution of the height of a priority queue file history with length $2n$. The schema of the history is the sequence of the successive sizes of the process. The problem has now become purely combinatorial.

More precisely, let $H_{2n}^{\leq h}$ denote the number of priority queue histories of length $2n$ and height $\leq h$. Since there are $1 \cdot 3 \cdots (2n - 1)$ priority queue histories of length $2n$ (just as many as there are involutions with no fixed point on $\{1, 2, \ldots, 2n\}$), we have

$$\Pr\left\{\max_{0 \leq t \leq 1}\{Need(t)\} \leq h\right\} = \frac{H_{2n}^{\leq h}}{1 \cdot 3 \cdot \cdots \cdot (2n - 1)}.$$

We can now use the results of Flajolet *et al.* [1980]:

$$H^{\leq h}(z) = \sum_n H_{2n}^{\leq h} z^{2n} = \frac{Q_{h-1}(z)}{Q_h(z)},$$

where $Q_h(z)$ is an even polynomial with degree h or $h + 1$ according to whether h is even or odd. Moreover, $H_{h+1}(z) = z^{h+1}Q_h(1/z)$ is the $(h + 1)$st orthogonal Hermite polynomial, whose roots are all real and distinct (see [Szegö, 1939] for the properties of orthogonal polynomials). Let $a_{1,h+1} < a_{2,h+1} < \cdots < a_{l,h+1}$ be the positive roots of $H_{h+1}(z)$. Thus

$$\frac{Q_{h-1}(z^{1/2})}{Q_h(z^{1/2})} = \sum_{1 \leq i \leq l} \sum_{k \geq 0} - a_{i,h+1}^{2k+1} \frac{Q_{h-1}(1/a_{i,h+1})}{Q_h'(1/a_{i,h+1})} z^k;$$

hence we have

$$\Pr\left\{\max_{0 \leq t \leq 1}\{Need(t)\} \leq h\right\} = 2^n \frac{n!}{(2n)!} \sum_{1 \leq i \leq l} - a_{i,h+1}^{2n+1} \frac{Q_{h-1}(1/a_{i,h+1})}{Q_h'(1/a_{i,h+1})},$$

and when n goes to infinity, this yields

$$\Pr\left\{\max_{0 \leq t \leq 1}\{Need(t)\} \leq h\right\} \sim - \frac{Q_{h-1}(1/a_{l,h+1})}{Q_h'(1/a_{l,h+1})} a_{l,h+1}^{2n+1} \frac{e^n}{\sqrt{2}} \frac{1}{(2n)^n}.$$

3. Optimal Bounds.   In this section we prove for the stationary $M/G/\infty$ model
that the experienced maximum storage needed (that is, the expected maximum
$M/G/\infty$ queue size) and the expected maximum storage used in excess of that
amount are within constant factors, respectively, of the expected storage needed and
wasted at any given time. The birth rate is a Poisson process with intensity λ. In the
special case of the $M/M/\infty$ model, the lifespans are given by the exponential
distribution with mean $1/\mu$. In the general $M/G/\infty$ model, the lifespan distribution
is arbitrary, with mean $1/\mu$. Our results hold in the asymptotic case, in which μ
and λ are sufficiently large; we assume that $\mu, \lambda \geq e^2 \approx 7.4$.
    The following two theorems are the main results of Section 3:

THEOREM 3.1.   *We have*

$$E\left(\max_{0 \leq t \leq 1} \{Need(t)\} \right) = O(E(Need)) = O\left(\frac{\lambda}{\mu} \right),$$

under the condition that $\mu = O(\lambda/\log \lambda)$ *in the* $M/M/\infty$ *case, and* $\mu = O(\lambda/\log^2 \lambda)$ *in*
the general $M/G/\infty$ *case.*

THEOREM 3.2.   *Let* $\varepsilon > 0$ *be any constant. Then if the number* H *of buckets in HwLD*
is $\Omega((\log \lambda)^{1+\varepsilon})$, *we have*

$$E\left(\max_{0 \leq t \leq 1} \{Use(t)\} - \max_{0 \leq t \leq 1} \{Need(t)\} \right) = O(E(Use - Need)) = O(H).$$

    The restrictions on μ and H in the theorems are extremely weak; they are
typically met in geometrical applications, for example [Van Wyk and Vitter, 1986].
In fact, it can be shown that Theorem 3.1 is not true if μ is too large; the restriction
is thus partly inherent in the problem. For Theorem 3.2, however, we conjecture
that the restriction $H = \Omega((\log \lambda)^{1+\varepsilon})$ can be lifted.
    We prove Theorem 3.1 in the next section and Theorem 3.2 in Section 3.2. Our
approach for both is to approximate the queueing process by a sequence of stages
of a discrete analog, which we call *time hashing*. The particular forms of time
hashing we use for the two cases are quite different. But they share the common
property that the early stages of the time hashing capture most of what is going on
in the queueing process; in the later stages the number of slots in the hash table
becomes smaller and smaller (and each slot covers a larger span of time) and the
contribution becomes less and less.
    It is interesting to compare the idea of time hashing with those of extendible
hashing and other methods of dynamic hashing, in which the reverse phenomenon
occurs: in the early stages, the slots are coarse, and as items are inserted, the
number of slots gets larger and larger [Fagin et al., 1979]. Time hashing bears no
relation to the hashing that gives its name to HwLD. In a sense, the concept of time
hashing is orthogonal to it, because the slots represent intervals of time, whereas in
HwLD an item born at a certain time can potentially be stored in any of the H

buckets. To make things clearer, we always use the terminology "bucket" when discussing the hashing inherent in HwLD and the term "slot" when discussing time hashing.

3.1. Maximum Size of the M/G/∞ Queue. This section is devoted to the proof of Theorem 3.1. The number H of buckets in the HwLD implementation does not affect the value of *Need* in any way, so we assume in this section that $H = 1$. The distribution of *Need(t)* is Poisson with mean λ/μ:

LEMMA 3.1. *For the M/G/∞ model we have*

$$\Pr\{Need(t) = i\} = \frac{e^{-\lambda/\mu}(\lambda/\mu)^i}{i!}.$$

PROOF. This is a well-known result for the M/M/∞ model; see [Feller, 1968], for example. It can also be shown for the more general M/G/∞ model by considering the nonstationary case and letting $t \to \infty$ [Kleinrock, 1975].   □

The proof of Theorem 3.1 relies on the following technique we introduce, called *time hashing*: Let K be an integer parameter to be specified later. We consider all items that are alive at some time during $[0, 1]$. Stages $k = 0, 1, 2, \ldots, K$ of time hashing are defined as follows: For $0 \le k \le K$, all items (intervals) that have lifespan in the range $((1/\mu)2^{k-1}, (1/\mu)2^k]$ and that are born in either the unit interval $(0, 1]$ or one of the end intervals $(-(1/\mu)2^k, 0]$ and $(1, \lceil \mu 2^{-k} \rceil (1/\mu)2^k]$ are put into stage k; in addition, for $k = 0$, the lifespan requirement is weakened so that the lifespan must be in the range $[0, 1/\mu]$. Each stage consists of a hash table of $\lceil \mu 2^{-k} \rceil + 1$ slots, as pictured in Figure 3.1. The jth slot, for $0 \le j \le \lceil \mu 2^{-k} \rceil$,
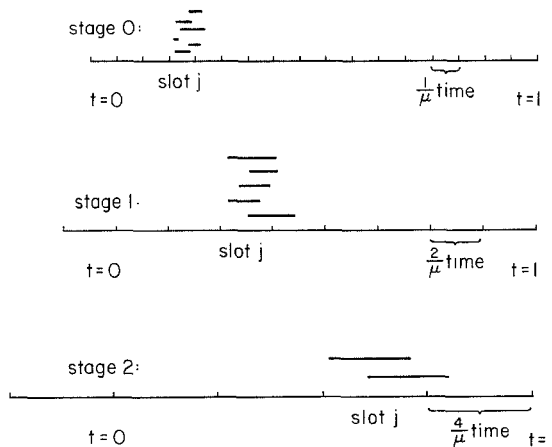

Fig. 3.1. Stages 0, 1, 2,... of time hashing, used to bound $E(\max_t\{Need(t)\})$. In the hash table for each stage, some typical items (intervals) belonging to slot j are pictured.

represents the interval of time $((1/\mu)(j - 1)2^k, (1/\mu)j2^k]$. An item in stage k is placed into the slot corresponding to its birthtime. We also define a special stage $K + 1$ as follows: slot 0 consists of all items born in $[0, 1]$ with lifespan $> (1/\mu)2^{K+1}$; the remaining $\lceil \mu 2^{-(K+1)} \rceil$ slots are left empty.

We define $N_k(j)$ to be the number of stage k items in slot j. The following fundamental relation bounds $\max_{0 \leq t \leq 1}\{Need(t)\}$ by the sum of the expected maximum slot occupancies in time hashing:

LEMMA 3.2. *We have*

$$\max_{0 \leq t \leq 1} \{Need(t)\} \leq 2 \sum_{0 \leq k \leq K+1} \max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\}.$$

PROOF. Let $Need_k(t)$ be the number of slots in stage k that are alive at time t:

$$\max_{0 \leq t \leq 1} \{Need(t)\} \leq \sum_{0 \leq k \leq K+1} \max_{0 \leq t \leq 1} \{Need_k(t)\}$$

$$\leq \sum_{0 \leq k \leq K} \max_{0 \leq j < \lceil \mu 2^{-k} \rceil} \{N_k(j) + N_k(j + 1)\} + 2N_{K+1}(0)$$

$$\leq 2 \sum_{0 \leq k \leq K+1} \max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\}.$$

The middle line of the derivation is based upon the fact that the only overlap possible between items in different slots in stage k, for $0 \leq k \leq K$, is between items in adjacent slots. For $k = K + 1$, it follows by symmetry that the total number of items with lifespan $> (1/\mu)2^{K+1}$ that are alive at some time in $[0, 1]$ is at most double the number of such items that are born in $[0, 1]$. The lemma follows immediately. ☐

The $M/M/\infty$ Case. First we handle the $M/M/\infty$ case, in which the lifetimes are exponentially distributed with mean $1/\mu$. The restriction on μ in Theorem 3.1 is slightly weaker in this case than in the general $M/G/\infty$ case. In this subsection we assume that we are dealing with the $M/M/\infty$ model and that $\mu = O(\lambda/\log \lambda)$. We define the stage parameter K to be $\lceil \lg \ln \mu \rceil$.

LEMMA 3.3. *The expected number of items in stage $K + 1$ is*

$$E(N_{K+1}(0)) = E\left(\max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_{K+1}(j)\} \right) \leq \frac{\lambda}{\mu}.$$

PROOF. An item is in stage $K + 1$ if it is born in $[0, 1]$ and if its lifespan is $> (1/\mu)2^{K+1}$. For the $M/M/\infty$ model the lifetimes are exponentially distributed with mean $1/\mu$, and the probability that an item has lifespan $> (1/\mu)2^{K+1}$ is $e^{-2^{K+1}}$. For our choice $K = \lceil \lg \ln \mu \rceil$, this probability is at most $1/\mu$. Hence, the average number of items in stage $K + 1$ is at most λ/μ. ☐

LEMMA 3.4.   *For $0 \leq k \leq K$, let n be the average number of items in stage k, and let $m = \lceil \mu 2^{-k} + 1 \rceil$ be the number of slots in the time hashing table of stage k. Then the number $N_k(j)$ of items in slot j of stage k is Poisson distributed with mean $\alpha = n/m$, where*

$$\alpha = \begin{cases} \dfrac{\lambda}{\mu} 2^k (e^{-2^{k-1}} - e^{-2^k}), & \text{if } 1 \leq k \leq K, \\[3mm] \dfrac{\lambda}{\mu} (1 - e^{-1}) & \text{if } k = 0. \end{cases}$$

PROOF.   This follows immediately from independence and from the fact that the birth rate is λ, the length of each slot is $(1/\mu)2^k$, the probability of a given item having lifespan in the range $[0, 1/\mu]$ is $1 - e^{-1}$, and the probability of having lifespan in the range $((1/\mu)2^{k-1}, (1/\mu)2^k]$ is $e^{-2^{k-1}} - e^{-2^k}$.   □

LEMMA 3.5.   *The expected maximum occupancy of the slots in stage k, $0 \leq k \leq K$, is*

$$E\left(\max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{ N_k(j) \} \right) = O\left(\frac{\lambda}{\mu 2^k} \right).$$

Before we prove Lemma 3.5, we turn our attention to the following lemma and corollary, which we use in the proof. They give us an upper bound in an easy way for the expected maximum slot occupancy in hashing. The lemma is phrased for general slot occupancies X_j that are not assumed to be independent; when the occupancies are independent or satisfy a certain property, the bound in the corollary is obtained. Similar lemmas have been used, for example, to show that the expected maximum occupancy in a hash table of load factor 1, under either the Bernoulli or Poisson distribution, is at most $2 \ln m / \ln \ln m(1 + o(1))$, where m is the number of table slots. The actual expected maximum occupancy is $\sim \ln m / \ln \ln m$, which was proved by Gonnet [1981] and Kolchin *et al.* [1978] by more complicated calculations.

LEMMA 3.6.   *For random variables $X_1, \ldots, X_m$, if $\Pr\{X_j > b\} \leq 1/(nm)$, for all $1 \leq j \leq m$, where $n = E(\sum_j X_j)$, then we have*

$$E\left(\max_{1 \leq j \leq m} \{ X_j \} \right) \leq b + \frac{1}{n} E\left(\max_{1 \leq j \leq m} \{ X_j \} \,\middle|\, \max_{1 \leq j \leq m} \{ X_j \} > b \right).$$

PROOF.   By hypothesis we have

$$\Pr\left\{ \max_{1 \leq j \leq m} \{ X_j \} > b \right\}$$

$$\leq \Pr\{X_1 > b\} + \Pr\{X_2 > b\} + \cdots + \Pr\{X_m > b\} \leq m \cdot \frac{1}{nm} = \frac{1}{n}.$$

The rest of the derivation consists of conditioning on whether $\max_{1 \le j \le m}\{X_j\}$ is $\le b$ or $> b$:

$$E\left(\max_{1 \le j \le m} \{X_j\}\right) \le b \cdot \Pr\left\{\max_{1 \le j \le m} \{X_j\} \le b\right\}$$

$$+ E\left(\max_{1 \le j \le m} \{X_j\} \,\middle|\, \max_{1 \le j \le m} \{X_j\} > b\right) \cdot \Pr\left\{\max_{1 \le j \le m} \{X_l\} > b\right\}$$

$$\le b + \frac{1}{n} E\left(\max_{1 \le j \le m} \{X_j\} \,\middle|\, \max_{1 \le j \le m} \{X_j\} > b\right). \qquad \square$$

The following corollary gives a bound that is easier to apply, at the cost of extra restrictions on the random variables $X_1, \ldots, X_m$. The restrictions are satisfied, for example, if the random variables are independent.

COROLLARY 3.1. *If in addition to the assumption required for Lemma 3.6 we also have*

$$E\left(\max_{1 \le j \le m} \{X_j\} \,\middle|\, \max_{1 \le j \le m} \{X_j\} > b\right) \le E\left(\max_{1 \le j \le m} \{X_j\} \,\middle|\, X_1 > b\right),$$

then

$$E\left(\max_{1 \le j \le m} \{X_j\}\right) \le b + \frac{1}{n} E\left(\max_{1 \le j \le m} \{X_j\} \,\middle|\, X_1 > b\right).$$

PROOF OF LEMMA 3.5. Suppose that $\mu \le c\lambda/\ln\lambda$, for $c \ge 2$. Let $d \ge 3$ be a constant to be specified later. We apply Corollary 3.1 to the random variables $X_j = N_k(j-1)$ with the parameters

$$b = \frac{d\lambda}{\mu 2^k},$$

$$n = \begin{cases} p\dfrac{\lceil \mu 2^{-k}\rceil + 1}{\mu 2^{-k}} (e^{-2^{k-1}} - e^{-2^k}) & \text{if } 1 \le k \le K, \\[3mm] \lambda\dfrac{\lceil \mu\rceil + 1}{\mu} (1 - e^{-1}) & \text{if } k = 0, \end{cases}$$

$$m = \lceil \mu 2^{-k}\rceil + 1.$$

First we must show that the hypotheses in Corollary 3.1 are satisfied. It is clear that the random variables $N_k(j)$ are independent. What remains is to show that the hypothesis carried over from Lemma 3.6 is true, namely, that

(3.1) $$\Pr\{N_k(j) > b\} \le \frac{1}{nm}.$$

By Lemma 3.4, $N_k(j)$ is a Poisson random variable with mean $\alpha = n/m$. We have

$$\Pr\{N_k(j) > b\} = e^{-\alpha} \sum_{i > b} \frac{\alpha^i}{i!} \leq 2e^{-\alpha} \frac{\alpha^b}{b!}.$$

By substituting the values of α and b and simplifying, we get the bound

$$(3.2) \qquad \Pr\{N_k(j) > b\} \leq 2\left(\left(\frac{4^k}{de^{2^{k-1}-1}}\right)^{1/2^k}\right)^{d\lambda/\mu} \leq 2(e^{-0.23})^{d\lambda/\mu}.$$

We also have $1/(nm) \geq 2/\lambda^2$. Combining this with (3.2), we can satisfy (3.1) if $(e^{-0.23})^{d\lambda/\mu} \leq 1/\lambda^2$. By taking logarithms and simplifying, we find that a sufficient condition is $\mu \leq \frac{1}{2}(0.23)d\lambda/\ln\lambda$. If we define $d = 8.7c$, then the sufficient condition becomes $\mu \leq 1.0005c\lambda/\ln\lambda$, which is satisfied by the hypothesis of Theorem 3.1, given at the beginning of this proof.

Now we can apply Corollary 3.1:

$$(3.3) \qquad E\left(\max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\}\right) \leq b + E\left(\max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\} \,\middle|\, N_k(0) > b\right).$$

The second term on the right-hand side of (3.3) is negligible and can be easily bounded: the values $N_k(j)$ are independent with respect to j. We have

$$(3.4) \qquad E\left(\max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\} \,\middle|\, N_k(0) > b\right)$$

$$\leq E\left(\max_{1 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\}\right) + E(N_k(0) \mid N_k(0) > b).$$

The first term on the right-hand side of (3.4) can be bounded coursely by the expected total number of items n in stage k. The second term is

$$\sum_{k > b} \frac{e^{-\alpha}\alpha^k}{k!} k \,\middle/\, \sum_{k > b} \frac{e^{-\alpha}\alpha^k}{k!} = O(b),$$

where $\alpha = n/m$ is the probability that a given item falls into slot 0. Substituting these bounds back into (3.4) and then into (3.3), we get

$$E\left(\max_{0 \leq j \leq \lceil \mu 2^{-k} \rceil} \{N_k(j)\}\right) = O(b) = O\left(\frac{\lambda}{\mu 2^k}\right).$$

This completes the proof of Lemma 3.5.   □

The rest of the proof of Theorem 3.1 for the M/M/∞ case consists of taking expectations in the expression of Lemma 3.2 and substituting the bounds from Lemmas 3.3 and 3.5, which gives a convergent geometric series.

The M/G/∞ Case. In this subsection we assume that $\mu = O(\lambda/\log^2 \lambda)$. For the case of the M/G/∞ model, the distribution of lifetimes is allowed to be an arbitrary one with mean $1/\mu$. So in particular the approach we used above for M/M/∞ (namely, Lemma 3.5) will not work; for each given value of k, stage k could contribute as much as $\Omega(\lambda/\mu)$ to $E(\max_{0 \le j \le \lceil \mu 2^{-k} \rceil}\{N_k(j)\})$. Instead we use the following important correspondence between the average slot occupancies and $E(Need)$:

LEMMA 3.7. *Let $\alpha_k = E(N_k(0))$ be the average number of items in slot 0 of stage k. Then*

$$\frac{1}{2} \sum_{1 \le k \le K+1} \alpha_k \le \frac{\lambda}{\mu}.$$

PROOF. An average of at most half the items in slot 0 are alive at time $t = 0$. The total number of items alive at time $t = 0$ is λ/μ. The lemma follows directly. □

We use time hashing as before, but with the stage parameter set to $K = \lceil \lg \mu \rceil$. There are $\lceil \mu 2^{-k} \rceil + 1 \le \mu + 2$ slots in stage k, for each $0 \le k \le K$. An easy application of Corollary 3.1 gives us the following:

LEMMA 3.8. *We have*

$$E\left(\max_{0 \le j \le \lceil \mu 2^{-k} \rceil} \{N_k(j)\}\right) = \begin{cases} O(\alpha_k) & \text{if } \alpha_k > \ln \mu, \\ O(\log \mu) & \text{if } \alpha_k \le \ln \mu. \end{cases}$$

Substituting Lemmas 3.3 and 3.8 into Lemma 3.2 and taking expectations, we get

$$(3.5) \qquad E\left(\max_{0 \le t \le 1} \{Need(t)\}\right) \le 2 \sum_{0 \le k \le K+1} E\left(\max_{0 \le j \le \lceil \mu 2^{-k} \rceil} \{N_k(j)\}\right)$$

$$= O\left(\sum_{0 \le k \le K} \alpha_k + \sum_{0 \le k \le K} \log \mu\right) + \frac{2\lambda}{\mu}$$

$$= O\left(\frac{\lambda}{\mu} + \log^2 \mu\right).$$

The last line in the derivation of (3.5) follows from Lemma 3.7 and the definition $K = \lceil \lg \mu \rceil$. Quantity (3.5) is $O(\lambda/\mu)$ when $\mu = O(\lambda/\log^2 \lambda)$. This completes the proof of Theorem 3.1 for the M/M/∞ case.

3.2. Optimal Bounds on the Waste in HwLD. To prove Theorem 3.2 we derive an upper bound for $E(\max_t\{Waste(t)\})$, where $Waste(t) = Use(t) - Need(t)$ is the number of dead items that are still in the HwLD data structure at time t. This therefore gives an upper bound on $E(\max_t\{Use(t)\} - \max_t\{Need(t)\})$. It is important to note that the former quantity is usually larger than the latter, because $Use(t)$ and $Need(t)$ typically do not attain their maxima at the same time t.
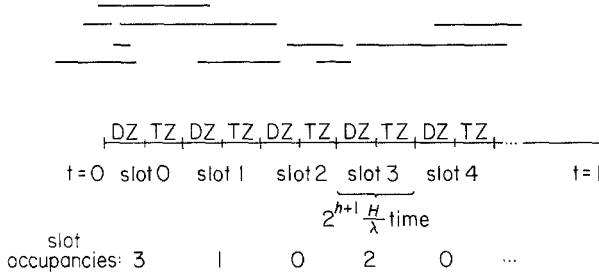
Fig. 3.2. An example of typical items stored in a particular bucket h in the HwLD data structure; the items are pictured above as intervals. The marks on the horizontal axis show the extents of the slots of the time hashing table for bucket h, stage k, for some $1 \le k \le K$; the death zones and twilight zones are denoted DZ and TZ, respectively. The numbers given below show the slot occupancy $w_{h,k}(j)$ for each slot, which is equal to the number of deaths in the slot's death zone, provided that there are no births in the twilight zone, or 0 otherwise. Note for example that there are no entries in slot 2 since there is a birth in the twilight zone.

To bound the expected maximum waste, we use a time hashing of a different nature than in Section 3.1. The stages are numbered $k = 0, 1, \ldots, K + 1$, and each of the H buckets has its own set of stages. The hash table for each bucket for stage k has $\lceil (\lambda/H)2^{-(k+1)} \rceil$ slots. The jth slot, for $0 \le j \le \lceil (\lambda/H)2^{-(k+1)} \rceil - 1$, represents the time interval $(j2^{k+1}(H/\lambda), (j + 1)2^{k+1}(H/\lambda)]$ and is shown in Figure 3.2. The first half of each slot is called the *death zone*, and the second half is called the *twilight zone*. For each stage, one entry is put into its jth slot for every death in the death zone of its jth slot, with the extra requirement that there are no births in the twilight zone of the jth slot; if there is a birth in the twilight zone, no entries are placed into the jth slot.

In addition, stages 0 and $K + 1$ are supplemented as follows: In stage 0 an entry is put into the jth slot for every death in the death zone, regardless of whether there have been no births in the twilight zone. In stage $K + 1$ we move all the entries into slot 0 from the other slots.

We let $w_{h,k}(j)$ denote the slot occupancy for the jth slot in the time hashing table for bucket h in the kth stage. The above description can be rephrased as

$$
w_{h,k}(j) = \begin{cases}
\begin{array}{l}\text{\# deaths in the death} \\ \text{zone of slot } j\end{array} & \text{if} \quad k = 0, \\[2ex]
\begin{array}{l}\text{\# deaths in the death} \\ \text{zone of slot } j\end{array} & \text{if} \quad 1 \le k \le K \quad \text{and} \quad \not\exists \text{ births in} \\ & \qquad \text{the twilight zone of slot } j, \\[2ex]
0 & \text{if} \quad 1 \le k \le K \quad \text{and} \quad \exists \text{ births in the} \\ & \qquad \text{twilight zone of slot } j, \\[2ex]
\begin{array}{l}\text{\# deaths in all death zones} \\ \text{for which } \not\exists \text{ births in} \\ \text{the twilight zones}\end{array} & \text{if} \quad k = K + 1 \quad \text{and} \quad j = 0, \\[3ex]
0 & \text{if} \quad k = K + 1 \quad \text{and} \quad j > 0.
\end{cases}
$$

We define $W_k(j)$ to be the total number of entries in the jth slots of the hash tables for buckets $1, 2, \ldots, H$:

$$W_k(j) = \sum_{1 \le h \le H} w_{h,k}(j).$$

We set the stage parameter K to be $K = \lceil \lg \ln(\lambda/H) \rceil$.

For completeness we should mention that there is a total of four instances of time hashing, not just the one defined above. The second instance of time hashing is defined in an identical way, except that the time intervals of the slots are offset $(H/\lambda)2^k$ from the time intervals of the instance defined above. In addition to these two instances, we consider two "reverse" instances, in which time is viewed backward: we start at time $t = 1$ and end at time $t = 0$, and we process each death as a birth and vice versa. Without loss of generality we discuss only the first instance of time hashing, as defined in the previous paragraphs, and introduce an extra factor of 4 into our bounds, where appropriate.

A key observation for the derivation is that the death rate in the $M/G/\infty$ model is a Poisson process with the same intensity as the birth rate. This follows because the $M/G/\infty$ model is symmetric and stationary, and thus also reversible [Kelly, 1979]. The following lemma is the basis for our proof of Theorem 3.2:

LEMMA 3.9. *We have*

$$\max_{0 \le t \le 1} \{Waste(t)\} \le 4 \sum_{0 \le k \le K+1} \max_{0 \le j \le \lceil (\lambda/H)2^{-(k+2)} \rceil} \{W_k(j)\}.$$

PROOF. Let us define the "waste" of an item to be the interval of time after its death that it remains in the HwLD data structure. Each waste interval in bucket h, for $1 \le h \le H$, can be decomposed into individual segments, such that each segment corresponds to an entry in one of the four instances of the time hashing, and the entry is in a slot whose time interval completely contains the segment's time interval. The factor of 4 appears because there are four instances of time hashing, as mentioned above, whereas the random variables $w_{h,k}(j)$ and $W_k(j)$ are defined for only the first instance. ☐

We prove Theorem 3.2 by bounding the sum in Lemma 3.9 by $O(E(Waste)) = O(H)$. A big difference between this application of time hashing and the ones we used in Section 3.1 is that the random variables $w_{h,k}(j)$ (and hence also $W_k(j)$) are almost always 0 as k grows. We have $\Pr\{w_{h,k}(j) = 0\} \approx 1 - e^{-2^k}$. This causes the maximum slot occupancy to behave wildly. In fact, to get our bound it is not enough to bound $E(\max_j\{w_{h,k}(j)\})$ and then multiply by H, because the result will be too large: the load factor in the analysis of $\max_j\{w_{h,k}(j)\}$ is too small, and the ratio between the average maximum slot occupancy and the average slot occupancy is no longer $O(1)$. The solution is to consider the H buckets *in toto* and to bound $E(\max_j\{W_k(j)\})$ directly. We do that by computing the moment generating function of $\max_{0 \le j \le \lceil (\lambda/H)2^{-(k+1)} \rceil}\{W_k(j)\}$ and then applying Corollary 3.1 using Chernoff's bound.

LEMMA 3.10. *The occupancy $w_{h,k}(j)$ of the jth slot in the time hashing table for the hth bucket and the kth stage, where $1 \le h \le H$, $0 \le k \le K+1$, $0 \le j \le \lceil (\lambda/H)2^{-(k+1)} \rceil - 1$, is distributed as follows:*

$$
\Pr\{w_{h,k}(j) = l\} = \begin{cases} e^{-\beta}\dfrac{e^{-\alpha}\alpha^l}{l!} & \text{if } l \ge 1, \\ 1 - e^{-\beta} + e^{-(\beta+\alpha)} & \text{if } l = 0, \end{cases}
$$

where

$$
\beta = \begin{cases} 0 & \text{if } k = 0, \\ 2^k & \text{if } 1 \le k \le K+1, \end{cases}
$$

$$
\alpha = \begin{cases} 2^k & \text{if } 0 \le k \le K, \\ 2^{K+1}\left\lceil \dfrac{\lambda}{H}2^{-(K+2)} \right\rceil & \text{if } k = K+1, \ j = 0, \\ 0 & \text{if } k = K+1, \ j > 0. \end{cases}
$$

PROOF. First let us consider stage k, where $1 \le k \le K$. For $l \ge 1$, the probability that there are l entries in slot j is the product of the probability $e^{-\alpha}\alpha^l/l!$ that there are l deaths in the death zone and the probability $e^{-\beta}$ that there are no births in the twilight zone. The probabilities for $l \ge 1$ determine the probability for $l = 0$. For stage $K+1$, the entries for the $\lceil (\lambda/H)2^{-(K+2)} \rceil$ slots are moved into slot 0, which increases α by a factor of $\lceil (\lambda/H)2^{-(K+2)} \rceil$ for $j = 0$ and makes α equal to 0 for the other slots. For stage 0, there is no ban on births in the twilight zone, so $\beta = 0$. □

The rest of the derivation of Theorem 3.2 consists of the following two lemmas:

LEMMA 3.11. *The expected number of entries in stage $K+1$ is*

$$
E(W_{K+1}(0)) = E\left(\max_{0 \le j \le \lceil (\lambda/H)2^{-(k+1)} \rceil} \{W_{K+1}(j)\} \right) = O(H).
$$

PROOF. It follows from Lemma 3.10 and the definition $K = \lceil \lg \ln(\lambda/H) \rceil$ that $E(w_{h,K+1}(0)) = O(1)$, and the lemma follows by summing on h. □

LEMMA 3.12. *The expected maximum occupancy of the slots in stage k, $0 \le k \le K$, is*

$$
E\left(\max_{0 \le j \le \lceil (\lambda/H)2^{-(k+1)} \rceil} \{W_k(j)\} \right) = O\left(\frac{H}{2^{k/d}} \right),
$$

if $H = \Omega((\log \lambda)^{1+1/d})$.

PROOF. Let us restrict ourselves to the harder case $1 \le k \le K$; the $k = 0$ case is similar. We apply Corollary 3.1 to the random variables $X_j = W_k(j-1)$ with the parameters

$$m = \left\lceil \frac{\lambda}{H} 2^{-(k+1)} \right\rceil,$$

(3.6)
$$n = mH2^k e^{-2^k},$$

$$b = \frac{H}{2^{k/d}},$$

where d is a positive constant. The random variables are independent, since the deaths form a Poisson process, as mentioned earlier. The remaining hypothesis of Corollary 3.1 to show is that

(3.7)
$$\Pr\{W_k(j) > b\} \le \frac{1}{nm}.$$

To do that, we use Chernoff's bound (see, for example, [Kleinrock, 1975])

(3.8)
$$\Pr\{X \ge u\} \le \frac{E(e^{sX})}{e^{su}},$$

where $s \ge 0$, and we consider the moment generating function $E(e^{sW_k(j)})$. By independence we have

(3.9)
$$E(e^{sW_k(j)}) = E(e^{s(w_{1,k}(j) + \cdots + w_{h,k}(j))}) = (E(e^{sw_{1,k}(j)}))^H.$$

By Lemma 3.10, we have

(3.10)
$$E(e^{sw_{1,k}(j)}) = \sum_{l \ge 0} e^{sl} \Pr\{w_{1,k}(j) = l\} = e^0 \Pr\{w_{1,k}(j) = 0\} + e^{-2^k} E(e^{sY}),$$

where Y is a Poisson random variable with mean 2^k. Its moment generation function is given by

(3.11)
$$E(e^{sY}) = e^{(e^s - 1)2^k}.$$

Combining (3.9) and (3.11) with Chernoff's bound (3.8), we get

(3.12)
$$\Pr\{W_k(j) > b\} \le \frac{(1 + e^{(e^s - 1)2^k})^H}{e^{sb}}.$$

We set $s = 0.5$. For some constant $c > 0$, the right-hand side of (3.12) is bounded by

(3.13)
$$O(e^{-H/(c2^{k/d})}).$$

The right-hand side of (3.7) can be bounded from below by $1/\lambda^2$. Combining this with (3.13) we find that (3.7) is satisfied if $H = \Omega((\log \lambda)^{1+1/d})$, which is true by assumption.

Now that the hypotheses of Corollary 3.1 are satisfied, we apply it and get

$$(3.14) \quad E\left(\max_{0 \leq j \leq \lceil (\lambda/H)2^{-(k+1)}\rceil} \{W_k(j)\}\right)$$

$$= O\left(\frac{H}{2^{k/d}} + \frac{1}{n} E\left(\max_{0 \leq j \leq \lceil (\lambda/H)2^{-(k+1)}\rceil} \{W_k(j)\} \,\middle|\, W_k(0) > \frac{H}{2^{k/d}}\right)\right).$$

Using Lemma 3.10, the same reasoning used in (3.4) gives us

$$E\left(\max_{0 \leq j \leq \lceil (\lambda/H)2^{-(k+1)}\rceil} \{W_k(j)\} \,\middle|\, W_k(0) > \frac{H}{2^{k/d}}\right) = O(H2^k),$$

and hence by (3.14) and (3.6) and the fact that $k \leq K = \lceil \lg \ln(\lambda/H)\rceil$, we get

$$E\left(\max_{0 \leq j \leq \lceil (\lambda/H)2^{-(k+1)}\rceil} \{W_k(j)\}\right) = O\left(\frac{H}{2^{k/d}} + \frac{H2^k e^{2^k}}{\lambda}\right)$$

$$= O\left(\frac{H}{2^{k/d}} + \log \frac{\lambda}{H}\right) = O\left(\frac{H}{2^{k/d}}\right).$$

This completes the proof of Lemma 3.12. $\square$

Theorem 3.2 follows by combining Lemmas 3.9, 3.11, and 3.12, and summing on k.

4. Conclusions. The maximum size attained by a queue over time is a basic notion in stochastic processes and queueing theory. In terms of data structures, if we model the insertions and deletions of elements as the birth and death of items in a queue, then the maximum queue size is the maximum size of the data structure. Our conclusions come in two forms: First, we have used in a natural way a variety of algebraic and analytical techniques to obtain exact formulas for the distribution of the maximum size of queues for birth-and-death processes and for hashing with lazy deletion (HwLD). Our solutions are amenable to numerical calculation and some asymptotics. The formulas for several different models are related in that the relevant transform in each case can be expressed as a ratio of classical orthogonal polynomials.

Second, we have answered some open questions in queueing theory using discrete, non-queueing-theory techniques. We have obtained optimal big-oh bounds on the expected maximum queue size for the M/G/∞ process and for HwLD. We prove for HwLD that the expected maximum amount of needed space (that is, the maximum size of the M/G/∞ queue, on the average) and the expected maximum amount of space used by HwLD above the optimal amount are within

small constant factors, respectively, of the average space needed and wasted at any given time. Our techniques also appear to be applicable to the M/M/1 model, which introduces several interesting new facets to the problem.

Current work is aimed at removing the condition $H = \Omega(\log \lambda)$ from Theorem 3.2. The proof technique, though, has to be different, because it is easy to show for $H = 1$ that $\max_{0 \leq t \leq 1}\{Waste(t)\}$ has unbounded expectation. Another problem being worked on is to determine the constant factors inherent in the big-oh bounds. Preliminary results suggest that the constants in Theorems 3.1 and 3.2 are asymptotically 1 under general conditions.

Addendum. Since this paper was submitted, new results have been obtained in which several of the big-oh results obtained in this paper have been shown to hold with asymptotic equality; that is, the constants implicit in the big-oh bounds are equal to 1 asymptotically [Kenyon and Vitter, 1991].

References

R. Fagin, J. Nievergelt, N. Pippenger, and H. R. Strong. Extendible Hashing—A Fast Access Method for Dynamic Files, *ACM Transactions on Database Systems*, 4(3) (September 1979), 315-344.

P. Flajolet. Analyse d'algorithmes de manipulation d'arbres et de fichiers, *Cahiers du Bureau Universitaire de Recherche Opérationnelle*, 34-35 (1981), 1-209.

P. Flajolet, J. Françon, and J. Vuillemin. Sequence of Operations Analysis for Dynamic Data Structures, *Journal of Algorithms*, 1(2) (June 1980), 111-141.

G. H. Gonnet. Expected Length of the Longest Probe Sequence in Hash Code Searching, *Journal of the ACM*, 28(2) (April 1981), 289-304.

S. Karlin and J. M. McGregor. Linear Growth Birth and Death Processes, *Journal of Mathematics and Mechanics*, 7(4) (1958).

F. P. Kelly. *Reversibility and Stochastic Networks*, Series in Probability and Mathematical Statistics, Wiley, Chichester (1979).

C. M. Kenyon and J. S. Vitter. General Methods for the Analysis of the Maximum Size of Dynamic Data Structures, *SIAM Journal on Computing*, 20(3) (June 1991).

L. Kleinrock, *Queueing Systems*, Vol. I, Wiley, New York (1975).

V. F. Kolchin, B. A. Sevast'yanov, and V. P. Chistyakov. *Random Allocations*, Winston, Washington (1978).

J. Morrison, L. A. Shepp, and C. J. Van Wyk. A Queueing Analysis of Hashing with Lazy Deletion, *SIAM Journal on Computing*, 16(6) (December 1987), 1155-1164.

T. Ottmann and D. Wood. Space-Economical Plane-Sweep Algorithms, *Computer Vision, Graphics, and Image Processing*, 34 (1986), 35-51.

G. Szegö. *Orthogonal Polynomials*, American Mathematical Society Colloquium Publication, Providence, RI (1939).

T. G. Szymanski and C. J. Van Wyk. Space-Efficient Algorithms for VLSI Artwork Analysis, *Proceedings of the 20th IEEE Design Automation Conference* (June 1983), pp. 743-749.

C. J. Van Wyk and J. S. Vitter. The Complexity of Hashing with Lazy Deletion, *Algorithmica*, 1(1) (March 1986), 17-29.