PUC

Series: Monografias em Ciência da Computação
Nº 22/77

EXPECTED LONGEST PROBE SEQUENCE IN HASH CODE SEARCHING

by

Gaston H. Gonnet

Departamento de Informática

Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vivente, 225 – ZC-19
Rio de Janeiro – Brasil

Series: Monografias em Ciência da Computação
Nº 22/77

Series Editor: Michael Challis December 1977

EXPECTED LONGEST PROBE SEQUENCE IN HASH CODE SEARCHING*

by

Gaston H. Gonnet

*This work was partially financed by FINEP

RESUMO

Definimos sequência máxima média, um caso pior médio, de uma tabela organizada com o método de "hash coding" ao valor médio do máximo número de acessos necessários para pesquisar qualquer elemento dentro de uma tabela aleatória. Esta definição é diferente da comum que é o caso pior para o pior arquivo.

Apresentamos expressões assintóticas destes valores médios para tabelas completas e parcialmente completas. Para o método de "open-addressing" os resultados são 0.6315... n e $\sim -log_{\alpha}(n)$ onde n é o número de chaves, m é o tamanho da tabela e $\alpha = n/m$. Para o algoritmo "open addressing", reordenando a inserção de forma a minimizar o caso pior encontramos cotas mínimas $ln(n) + 1.077...$ $\lceil -\alpha^{-1} ln(1 - \alpha) \rceil$ respectivamente. Finalmente para o algoritmo "separate chaining" ("direct chaining") ambas médias são $\sim \Gamma^{-1}(m)$.

Estes resultados indicam que o comportamento real do caso pior em média é bom.

Palavras chave: Hashing; análise de algoritmos, caso pior; "open addressing" separate chaining; análise assintótica; hashing ótimo; hashing minimax; valor médio.

Abstract.

We define the expected longest probe sequence, an expected worst case, of a hash table as the expected value of the maximum number of accesses needed to locate any element in a random file. This differs from the usual definition of worst case which, in the case of hashing, is the longest sequence of accesses for the worst file.

We find asymptotic expressions of these expected values for full and partially filled tables. For the open addressing scheme with a clustering-free model we find these values to be 0.6315 ... n and $\sim -log_{\alpha}(n)$, where n is the number of records, m is the size of the table, and $\alpha = n/m$. For the open addressing scheme reordering the insertion such that we minimize the worst case, under a random probing model, we find the tight lower bounds $ln(n) + 1.077...$ and $\lceil -\alpha^{-1} ln(1 - \alpha) \rceil$ respectively. Finally for the separate chaining (or direct chaining) method we find both expected values to be $\sim \Gamma^{-1}(n)$.

These results show that generally the actual behaviour of the worst case in hash tables is on average quite good.

Keywords. Hashing; analysis of algorithms; worst case; open addressing; separate chaining, direct chaining; asymptotic analysis; optimal hashing; minimax hashing; expected value; average case.

1. INTRODUCTION

Generally we can divide the sorks in hash coding methods into three groups: how to compute good hashing functions, new algorithms to resolve collisions, and the analysis of these algorithms under various models of keys. This paper is focused on the last area, in this case, the analysis of the average longest probe sequence in a hashing table.

It is well known that a hash table, when we insert n keys, may have a worst case of n accesses to insert/locate an element, that is, for some key (the last), we may need up to n probes. Note however, that this is the worst case of the worst possible table. For some hashing schemes, this worst case occurs with a ridiculously small probability. This result, besides being not very encouraging, does not contribute any information about what we should expect as the longest sequence in a random file. In other words, the length of the longest probe sequence of a random file is a random variable; we know that this random variable has a maximum value of n, but its average value, i.e. the average longest probe sequence, is also of much interest.

In this paper we find asymptotic expressions for full and partially filled tables for three hashing schemes: open-addressing with a clustering-free hash function (uniform probing), a reordering scheme for open-addressing which minimizes the longest probe sequence, and separate (or direct) chaining.

In the next section we present a table of results on the average longest probe sequence with the average number of accesses for several algorithms. Sections 3, 4 and 5 are more mathematically oriented and give the derivation of the new results for each of the hashing schemes analyzed. Finally the last section presents the conclusions, consequences, and remaining work in this area.

2. THE RESULTS

Table I summarizes the new results together with results for the average number of accesses. For completeness we present various hashing schemes.

Algorithm	Average Number of Accesses		Average Longest Probe Sequence	
	Full table	$\alpha = n/m$	Full table	$\alpha = n/m$
Open Address. (uniform probing) [Knu 6.4.D]	$ln(m) - 1 + \gamma + o(1)$	$-\alpha^{-1} ln(1-\alpha)$	$0.63158... \times m + O(1)$	$-log(m) + O(log_\alpha(-log_\alpha(m)))$
Brent's reordering scheme [Bre]	$2.49 ...$	*complicated*	$O(\sqrt{m})$ (c)	?
Gonnet & Munro reordering scheme [Gon 2]	$2.13 ...$	*complicated*	$O(ln\ m)$ (c)	?
Optimal reordering to minimize average [Gon 2]	$\geq 1.668...$ [Gon 1] ~ 1.83 (e)	$\geq 2 - \frac{1-e^{-\alpha}}{\alpha}$	$O(ln\ m)$ (e)	?
Optimal reor. to minimize worst case [Gon 2]	~ 1.83 (e)	?	$ln(m) + 1.077 ... o(1)$	$\lceil -\alpha^{-1} ln(1-\alpha) \rceil$
Separate chaining [Knu 6.4]	1.5	$1 + \alpha/2$	$\sim \Gamma^{-1}(m)$	$\sim \Gamma^{-1}(m)$

Table I

The results marked with (c) are conjectured values, the ones marked with (e) are experimental results found by simulation; "complicated" in these cases indicates that results were only obtained by numerical integration of systems of differential equations.

The first row shows the results for open-addressing [Knu 6.4]. This method resolves collisions computing new probe positions until all the

table is searched. We use for the analysis a clustering-free model (uniform probing), that is, each key probes the table in an independent random permutation of positions. The fact that the result for full tables is $O(n)$ is not surprising or new (the last element takes $(n+1)/2$ probes on the average), however the value of the constant factor, 0.63158..., is new. The result for partially filled tables is quite surprising since (for a fixed α) an $O(\log n)$ expected worst element behaviour is satisfactory.

Brent's algorithm is essentially a reordering of keys during insertion. If the key to be inserted probes to an occupied position, then the next probe positions of this key are also scanned.

Gonnet & Munro's algorithm is also a reordering scheme during insertion. It searches breadth first for the first empty location in the binary tree obtained from the probe sequence of the original key and those of each key in occupied locations.

The fourth row shows results obtained from the optimal reordering during insertion (an assignment problem) that minimizes the average number of accesses. These results are interesting since they provide lower bounds for all possible reordering schemes using open-addressing.

Similarly, the optimal reordering of insertion to minimize the longest probe sequence (also an assignment problem) provides lower bounds on the average longest sequence for any open-addressing scheme. These later lower bounds [Gon 2] proved to be tight. For full tables the results show that we may do even better than binary search for the longest probe sequence, since we have $\ln(n)$ rather than $\log_2(n)$, while the $O(1)$ for the average number of accesses is preserved. For partially filled tables we find an integer (depending only on α) coming from a very familiar formula, i.e. the average number of accesses for open-addressing.

The last row shows the results for the separate (or direct) chaining technique. This method forms linked lists with all keys that hash to the same location. For any α the average longest probe sequence depends on the inverse of the gamma function (the inverse of the factorial). This function grows very slowly (less than $\log n$) and hence shows another good property of separate chaining hashing.

3. AVERAGE MAXIMUM PROBE SEQUENCE FOR OPEN-ADDRESSING

In this section we will use a clustering-free model of a hashing function. That is, each key will probe the table in a sequence of positions that is a random permutation of all table entries. This model is also called uniform probing. The work by Guibas [Gui 1] and Guibas & Szemeredi [Gui 2] shows that for practical purposes second or higher order clustering is equivalent to uniform probing.

Full Tables

It is well known that the average longest probe sequence of a full table organized with the scheme just mentioned is $O(n)$ (the last key requires $(n + 1)/2$ probes on the average). Consequently our next interest is to find the factor that multiplies n for the longest probe sequence.

Considering the last elements inserted we find:

$$Pr\{k\text{-last key requires} \leq j \text{ probes}\} = 1 - \frac{(m - j)_{(k)}}{m_{(k)}}$$

$$= 1 - y^k + O(1/m) \tag{3.1}$$

where $y = (m - j)/m$, the parenthesized subscript k indicates descending factorial and the 1-last is the last, the 2-last is the next to last, etc.

Consequently considering the maximum probe sequence among the k last keys, which have all independent paths we derive:

$$Pr\{\text{longest sequence in last } k \text{ keys} \leq j\} = \prod_{i=1}^{k} (1 - y^i) + O(1/m) ,$$

$$E\{\text{longest sequence in last } k \text{ keys}\} = \sum_{j=0}^{m} \{1 - \prod_{i=1}^{k} (1 - y^i)\} + O(1)$$

$$= m\{1 - \int_0^1 \sum_{i=1}^{k} (1 - y^i) dy\} + O(1) . \tag{3.2}$$

For the latter step we bound the difference between the sum and the integral using the fact that the integrand has total variation 1, and change y for $1-y$.

Using the famous Euler identity related to partitions:

$$\prod_{i=1}^{\infty} (1 - x^i) = \sum_{i=-\infty}^{\infty} (-1)^i \, x^{(3i^2+i)/2} \qquad \text{[Abr 24.2.1]}$$

we obtain the limit of the sequences of constants when $k \to \infty$, i.e.

Theorem

$$E(\text{longest sequence}) = m\left\{1 - \sum_{i=-\infty}^{\infty} \frac{2(-1)^i}{3i^2 + i + 2}\right\} + O(1)$$

$$= 0.631587464\ldots\ m + O(1)\ . \qquad (3.3)$$

The next table shows the exact expected values for some table sizes, rounded to 4 decimal places derived from the exact formula.

m	E(long.seq.)	E(long.seq.)/m
5	3.3696	0.67392
10	6.5226	0.65226
40	25.4675	0.63669
100	63.3624	0.63362
∞	∞	0.631587464...

Table II

Partially Filled Tables

When the hash table is partially filled with n entries, to an $\alpha = n/m$ occupation factor, the probability of needing k or less accesses when inserting the next key is:

$$Pr\{\text{needing} \leq k \text{ accesses}\} = 1 - \frac{n_{(k)}}{m_{(k)}}$$

The probability that the longest sequence, when inserting the first n keys, is no longer than k is:

$$Pr\{longest\ sequence \leq k\} = \prod_{i=0}^{n-1} (1 - i_{(k)}/m_{(k)}) \tag{3.4}$$

$$ln(Pr\{longest\ sequence \leq k\}) = \sum_{i=0}^{n-1} ln(1 - i_{(k)}/m_{(k)})$$

$$= - \sum_{i=0}^{n-1} \left[\frac{i_{(k)}}{m_{(k)}} + \frac{1}{2} \left(\frac{i_{(k)}}{m_{(k)}}\right)^2 + \frac{1}{3} \cdots \right]$$

The median of the distribution, when α is fixed and $n,m \to \infty$ is given by j in:

$$ln(\tfrac{1}{2}) = \frac{-n_{(j+1)}}{(j+1)\, m_{(j)}} - \frac{1}{2} \cdots$$

and for small j

$$ln(\tfrac{1}{2}) = - \frac{n\alpha^j}{j+1} + O(n\alpha^{2j}) = - \frac{m\alpha^{j+1}}{j+1} + O(n)^{-1}. \tag{3.5}$$

Let $w(x)$ be the solution of the transcendental equation $w(x)\, e^{w(x)} = x$, and $w(x) = ln(x) - ln(ln(x)) + o(1)$ then

$$j = \frac{w(-m\ log_2 \alpha)}{-ln(\alpha)} - 1 + o(1) \tag{3.6}$$

To complete the proof, we need to show that the expected value is asymptotically equivalent to the median, i.e.

$$\lim_{m \to \infty} \frac{E(longest\ sequence)}{j} = 1$$

The expected value is defined by

$$E(\text{longest sequence}) = \sum_{k=0}^{\infty} (1 - Pr\{\text{longest sequence} \leq k\}) \ . \qquad (3.7)$$

We first analyze the sum up to the term $j - ln(j)$; for each of the terms using (3.5) we have:

$$ln(Pr\{long.seq. \leq k\}) < -\frac{m\alpha^{k+1}}{k+1} \leq \frac{m\alpha^{(j - ln(j) + 1)}}{j - ln(j) + 1} \sim$$

$$\sim -\frac{ln(2)(j+1)}{j+1-ln(j)} \ \alpha^{-ln(j)} < -\ ln(2)\ j^{-ln\,\alpha} \ .$$

The corresponding sum of those terms, which is a partial sum of $E(long.\ seq.)$, is greater than

$$[j - ln(j)]\cdot[1 - (1/2)^{j^{-ln(\alpha)}}] < E(long.seq.)$$

When $n \to \infty$ (3.6) implies that $j \to \infty$ and we first conclude that:

$$\lim_{n\to\infty} \frac{E(long.seq.)}{j} \geq 1 \ .$$

The contribution to $E(long.seq.)$ by terms with $k \geq j$ is approximated by:

$$\sum_{k\geq j} (1 - e^{ln\ Pr\{long.seq. \leq k\}}) \leq \sum_{k\geq j} (1 - e^{\frac{-m\alpha^{k+1}}{k+1}}) < \sum_{k\geq j} (1 - 1 + \frac{m\alpha^{k+1}}{k+1})$$

$$< m\frac{\alpha^{j+1}}{(1-\alpha)(j+1)} < \frac{ln(2)}{1-\alpha}$$

so we conclude that

$$E(long.seq.) < j + \frac{ln(2)}{1-\alpha} \qquad \text{and} \qquad \lim_{n\to\infty} \frac{E(long.seq.)}{j} \leq 1$$

and finally

Theorem

$$\lim_{n\to\infty} \frac{E(long.seq.)}{j} = 1$$

i.e.

$$E(long.seq.) \sim \frac{w(-m\ log_2(\alpha))}{-ln(\alpha)} - 1 \sim -\log_\alpha m - \log_\alpha(-\log_\alpha m) + O(1)$$

The following table shows some exact values of $E(long.seq.)$ using formulas (3.4) and (3.7) rounded to 4 decimal places and, in brackets, the value of the median j (3.6) rounded to 2 decimal places. Note that for all the values in the table the log-log term gives a significant contribution

α \ m	20	100	1000	10000	10^6
80%	5.4733 (5.61)	10.8611 (10.38)	19.4494 (18.33)	(26.99)	(45.36)
90%	7.6583 (9.03)	17.8671 (18.16)	35.7466 (34.23)	(52.18)	(90.72)
95%	9.5475 (13.04)	26.7479 (29.38)	(60.51)	(96.43)	(174.72)
99%	N/A	48.2629 (69.81)	(196.74)	(364.67)	(751.13)

Table III

4. AVERAGE LOWER BOUND FOR THE LONGEST PROBE SEQUENCE

The minimax hash coding problem is defined as follows: given a set of keys and a hashing function, find the order in which keys have to be inserted so that the maximum number of accesses to locate any single element is minimized. This minimum maximum number of accesses will be called the minimax value. A lower bound on the minimax is also a lower bound on the average longest sequence of any other open-addressing scheme.

We will first consider the case of a full table, i.e., $\alpha = 1$. Our model for this analysis will require that the hashing function produces for any key, an independent random sequence of probes from a discrete rectangular distribution in $(1,m)$. This is slightly different from the usual hashing functions since we allow probe positions to be repeated.

A necessary, but not sufficient, condition to generate a hashing table for $n = m$ with at most k accesses is that for each position i from 1 to m, there must be some key such that i is among the first k probe positions of that key. For each table, the smallest k satisfying the above condition is a lower bound on the corresponding minimax value. Consequently

$$E(minimax) \geq E(smallest\ k) \ .$$

Given k, the probability of all the table positions $(1,m)$ appearing among $k.m$ probe position is an occupancy distribution, also known as Arfwedson's distribution [Arf, Ste, Joh, Fel] denoted by $A_m(k.m)$

$$A_m(k.m) = \sum_{i=0}^{n} (-1)^i \binom{m}{i} (1 - i/m)^{km}$$

$$= m!\ m^{-km} \left\{ {km \atop m} \right\} \sim e^{-me^{-k}} \qquad (4.1)$$

the latter approximation being described in Feller (pp. 93-94), and the braces denoting Stirling numbers of the second kind [Abr].

The expected value of the smallest k, i.e. the average lower bound for the above distribution is

$$E(\text{smallest } k) = \sum_{k=1}^{\infty} k[A_m(k.m) - A_m((k-1.m)]$$

$$= \sum_{k=0}^{\infty} (1 - A_m(k.m)) \sim \sum_{k=0}^{\infty} (1-e^{-me^{-k}}) = Q(m) \qquad (4.2)$$

By inspection of the last summation we find the functional equation,

$$Q(m) = Q(me) - 1 + e^{-m} .$$

The general solution of the above equation is:

$$Q(m) = \ln(m) + 1.07721566 \ldots + P(\ln(m)) + O(e^{-m}) , \qquad (4.3)$$

where $P(x)$ is a periodic function with period 1 and

$$|P(x)| \leq 0.0001035 .$$

A much more complicated approach using complex integration (similar to Knu 5.2.2) shows that the constant in 4.3 is $\gamma + 1/2$. For practical purposes, ignoring the periodic contribution, we have the

Theorem

$$\text{lower bound} = E(\text{smallest}k) = \ln(m) + 1.077 \ldots + o(1) . \qquad (4.4)$$

Table IV shows the exact lower bounds computed from (4.2), rounded to 4 decimal places together with the asymptotic value, computed from (4.4), in brackets.

m	E(lower bound)
5	2.6956/(2.6867)
10	3.3761/(3.3798)
20	4.0737/(4.0729)
50	4.9892/(4.9892)
1000	(7.9850)

Table IV

When we do not have a full table, we define T_k to be the number (a random variable) of different table positions that appeared in the first k probe positions of the n keys. If $T_k < n$ we cannot construct the table with at most k accesses. If $T_k \geq n$ it may be possible to construct the table with k accesses, that is

$$Pr\{not\ succeeding\} \geq Pr\{T_k < n\}$$

The distribution of T_k is also an occupancy distribution (Ste, Joh) and

$$E(T_k) = m(1 - (1 - 1/m)^{nk}) = m(1 - e^{k\alpha}) + O(1) ,$$

$$var(T_k) = m(1 - 1/m)^{nk} + m(m - 1)(1 - 2/m)^{nk} - m^2(1 - 1/m)^{2nk}$$

$$= m(e^{-k\alpha} - (1 + \alpha^2)e^{-2k\alpha}) + O(1) ,$$

and

$$coef\text{-}var(T_k) = O(m^{-1/2}) .$$

Consequently for $n,m \to \infty$

$$\begin{aligned} Pr\{T_k \geq n\} &\to 0 \quad iff\ E(T_k) < n \\ &\to 1 \quad iff\ E(T_k) > n , \end{aligned}$$

(note that $E(T_k)$ cannot be an integer for $nk > 1$) and the expected value of the lower bound on the minimax is the smallest integer k such that

$$E(T_k) > n \Rightarrow m(1 - e^{-k\alpha}) > n$$

which is $\lceil -\alpha^{-1} \ln(1 - \alpha) \rceil$. Note that the distribution is single valued when $n,m \to \infty$. Thus we have

Theorem

$$E(low\text{-}bound) = \lceil -\alpha^{-1} \ln(1 - \alpha) \rceil ,$$

with variance 0.

Table V shows the limits for the occupation factor α, that correspond to an expected lower bound k.

upper limit for α	E(lower bound)
0.7968	2
0.94048	3
0.98017	4
0.993023	5
0.997484	6
0.9996636	8
0.99995458	10

Table V

5. SEPARATE CHAINING

The technique of separate chaining, also called direct chaining or separate overflow chaining, forms a linked list of all keys that hash to the same location. The number of keys that will hash to a given location, i.e. the length of the linked list, has a probability distribution:

$$Pr\{list\ of\ length\ k\} = \binom{n}{k} m^{-n}(m-1)^{n-k} \sim \frac{e^{-\alpha}\alpha^k}{k!}$$

where $\alpha = n/m$ is the load factor. The last term is the Poisson approximation to the binomial distribution, that for a fixed α is very accurate. Consequently we will consider our model of separate chaining as having chains with length Poisson-independent distributed.

Let $e_i(\alpha)$ be the cumulative distribution of the above i.e.

$$e_i(\alpha) = \sum_{k=0}^{i} \frac{e^{-\alpha}\alpha^k}{k!} ,$$

and $d_i(\alpha) = 1 - e_i(\alpha)$. We then derive

$$Pr\{longest\ probe\ sequence \leq i\} = e_i(\alpha)^m ,$$

and

$$E(long.\ seq.) = \sum_{i=0}^{\infty} (1 - e_i(\alpha)^m) . \tag{5.1}$$

The median j is given by $\frac{1}{2} = e_j(\alpha)^m = (1 - d_j(\alpha))^m$, or taking logarithms and expanding $-ln(2) = -m(d_j(\alpha) + d_j(\alpha)^2/2 + \ldots)$. Clearly

$$d_j(\alpha) = O(m^{-1}) ,$$

and

$$\frac{ln(2)}{m} = d_j(\alpha) + O(m^{-2}) .$$

From the definition of $d_j(\alpha)$ it follows that

$$ln(2) = m \frac{e^{-\alpha}\ \alpha^{j+1}}{(j+1)!} + O(1/j) + O(1/m) .$$

For $\alpha = 1$ we conclude that

Theorem

$$j \sim \Gamma^{-1}\left[\frac{m}{e\ ln(2)}\right] - 2 = \Gamma^{-1}(m) + O(1)$$

when $\alpha \neq 1$ we also have

$$\frac{\Gamma^{-1}(m)}{j} = 1 + O(1/ln(j)) ,$$

but, as the asymptotic error term indicates, this limit is reached much more slowly than for $\alpha = 1$ for reasonable size files.

Following the same line of argument as in section 3 for partially filled tables, we can prove that the median asymptotically coincides with the expected value i.e.

$$\lim_{m\to\infty} \frac{E(long.seq.)}{j} = 1$$

Consequently

$$E(long.seq.) \sim \Gamma^{-1}(m)$$

The following table shows some exact expected longest probe sequences computed from (5.1) and the median j in brackets, rounded to 4 decimal places.

m \ α	1/2	1	2
24	2.2360	3.3347/(3.5650)	5.1688
120	3.0363	4.3359/(4.6209)	6.4423
720	3.8444	5.3432/(5.6568)	7.7055
5040	4.6541	6.3493/(6.6823)	8.9519

Table VI

6. CONCLUSIONS

Normally the worst case of hash coding is simply considered to be n. This statement implies a double worst case, i.e. the worst key for the worst file.

The longest probe sequence for a random file is a random variable (whose maximum value is n), with an expected value of much greater interest. It gives an idea of what we may expect to be the longest probe sequence for a given file.

Except for full tables in simple open addressing, these expected longest sequences are very slow growing functions (logarithm and inverse factorial). For some critical applications we may use the minimax reordering scheme, hence generally obtaining a $O(1)$ behaviour for the average, and a worst case better than binary search. If we can afford the use of pointers in the table, these results prove another good property of separate chaining.

The results are asymptotic, i.e. valid when $n,m \to \infty$, however the computed tables show that these are close approximations for reasonable size tables.

REFERENCES

[Abr] Abramowitz, M., and Stegun, I. A., Handbook of Mathematical Functions. Dover Publications, New York, 1964.

[Arf] Arfwedson, G., A Probability Distribution Connected with Stirling's Second Class Numbers. Skandinavisk Aktuarietidskrift, 34-3 (1951), 121-132.

[Bre] Brent, R. P., Reducing the Retrieval Time of Scatter Storage Techniques, CACM 16, 2 (Feb. 1973), pp. 105-109.

[Fel] Feller, W., An Introduction to Probability Theory and its Applications, John Wiley, New York, 1957, Vol. I.

[Gon 1] Gonnet, G. H., Average Lower Bounds for Open-Addressing Hash Coding, Proceedings of the Conference on Theoretical Computer Science, University of Waterloo, Waterloo, Ontario, Canada (Aug. 1977), pp. 159-162.

[Gon 2] Gonnet, G. H., and Munro, J. I., The Analysis of an Improved Hashing Technique, Proceedings of the Ninth Annual ACM Symposium on the Theory of Computing. Boulder, Colorado (May 1977).

[Gui 1] Guibas, L. J., The Analysis of Hashing Algorithms that Exhibit k-ary Clustering, Proceedings 17th Annual IEEE-FOCS Symp. Houston Texas (Oct. 1976), pp. 183-196.

[Gui 2] Guibas, L. J., and Szemeredi, E., The Analysis of Double Hashing, Proceedings of the 8th ACM Symposium on the Theory of Computing, Hershey, Pennsylvania (May 1967), pp. 187-191.

[Joh] Johnson, N. L., and Kotz, S., Distributions in Statistics, Houghton Mifflin, Boston, 1969, Vol. I (Discrete Distributions).

[Knu] Knuth, D. E., The Art of Computer Programming, Addison-Wesley, Don Mills (1973) Vol. III, Sorting and Searching.

[Riv] Rivest, R. L., Optimal Arrangement of Keys in a Hash Table, to appear in JACM.

[Ste] Stevens, W. L., Significance Grouping, Annals of Eugenics, 8 (1937), 57-69.