Full table quadratic quotient searching*

W. A. Burkhard

Computer Science Division, Department of Applied Physics and Information Science, University of California, San Diego, La Jolla, California 92037, USA

A scatter table search technique incorporating methods of the quadratic quotient search as well as the full table quadratic search is presented. The advantages of both techniques are retained. For table sizes a prime of the form $4j + 3$, the full table quadratic quotient search can access the entire scatter table via a computationally simple technique. Both primary and secondary clustering are avoided as well. Simulation results are presented for several of the search techniques.

(Received December 1973)

1. Introduction

Scatter storage is a well-known technique for implementing tables which allow rapid insertion, inspection, and deletion of items in the table. Two aspects of scatter storage are central here. First, there is the notion of transforming (scattering) the item into an initial address within the table. Many methods are available to do this; a common approach is to divide the item by the size of the table and use the remainder as the relative table address. Second, there is the choice of actions to be followed in the event that at least two items have the same initial address. These actions are referred to as search methods. This paper presents a new search method which is referred to as the **full table quadratic quotient** method, or as the FTQQ method.

The current literature contains many search techniques (Morris, 1968; Maurer, 1968; Bell, 1970; Bell and Kaman, 1970; Radke, 1970; Day, 1970; Hopgood and Davenport, 1972; Ecker, 1974) which represent substantial improvement over the original scatter storage strategy (Dumey, 1956; Peterson, 1957). One approach views the table as a directory. Entries in the table are addresses of chains of items (stored elsewhere in memory); each chain contains only those items with the same initial table address. At the expense of additional memory, this approach offers rapid insertion, inspection, and deletion of items in the table (Morris, 1968). When additional memory is not available, a significant improvement in the performance of the scatter storage techniques is obtained using quadratic search (Maurer, 1968). The quadratic search method has been improved in several ways.

The first refinement is concerned with searches which follow through identical (sub) sequences of addresses within the table. In particular, those searches which begin at the same table address and then follow through identical sequences of addresses within the table. This effect is referred to as secondary clustering. One approach to avoiding this type of clustering is presented in Bell (1970). The same ideas have been used in conjunction with linear search to avoid secondary clustering there as well (Bell and Kaman, 1970).

The second refinement is concerned with the fraction of the table accessible during any particular search. It is possible to modify the quadratic search in several ways such that the fraction of the table accessible during any particular search is changed from approximately 1/2 to 1. One approach is to use a 'double' quadratic search (Radke, 1970) and the technique has been further refined in Day, (1970) to obtain a computationally simple algorithm for a full table quadratic search of tables of size p where p is a prime of the form $4j + 3$ for some natural number j. Another approach to the quadratic search is presented in Hopgood and Davenport (1972) which allows a full table search for tables of size 2^α for $\alpha \geq 1$. A generalisation of the quadratic search method is presented in Ecker (1974) and upper and lower bounds for the search period are given as well. Moreover, it is shown that for many table sizes other than a prime, the entire table may be searched.

This paper is concerned with a scatter table search technique which combines ideas of Bell, (1970); Bell and Kaman, (1970); Radke, (1970); and Day, (1970) with quadratic search while retaining all the advantages of ancestors of the technique. Simulation results are presented for several of the search techniques.

2. The new search method

The full table quadratic quotient search method is presented as two routines, SCATTER and RESOLVE. The routine SCATTER is invoked knowing only the item for which a search is to be conducted and determines the initial address for the search. The routine RESOLVE is invoked as needed to generate additional addresses to continue the search for the item. The item is referred to as KEY and the table size is referred to as SIZE which must be a prime of the form $4j + 3$ where j is a natural number.

SCATTER:

Step 1. ADDRESS ← KEY MOD SIZE
Q ← (KEY/SIZE) MOD SIZE

Step 2. IF Q = 0 THEN
Q ← IF ADDRESS = 0 THEN 1 ELSE ADDRESS

Step 3. INDEX ← −Q * SIZE

RESOLVE:

Step 1. INDEX ← INDEX + 2 × Q
IF INDEX = SIZE × Q THEN
'all table addresses have been generated—set flag accordingly'

Step 2. ADDRESS ← (ADDRESS + |INDEX|) MOD SIZE

3. Analysis

This section of the paper is devoted to demonstrating that (*a*) the full table quadratic quotient search method does access the entire table and (*b*) that the search method does avoid secondary clustering. The investigation of secondary clustering shows another interesting structural property of the set of sequences of addresses generated by the method.

Let us begin by assuming that the table size is p, a prime of the form $4j + 3$ where j is a natural number and that Z_p denotes the integers mod p. It is apparent that the approach generates a sequence of differences of addresses equivalent mod p to the following sequence:

$$(p-2)Q, (p-4)Q, \ldots, 5Q, 3Q, Q, Q, 3Q, 5Q \ldots, (p-4)Q, (p-2)Q \quad (1)$$

*This work was supported in part by the National Science Foundation Grant GJ-34655 and UCSD Committee on Research.

The variable ABS(INDEX) generates this sequence exactly. Thus, the function $f: Z_p \times Z_p \times Z_p \to Z_p$ specified as follows:

$$f(i, Q, R) \equiv \begin{cases} R - Qi(i+1) & \text{if } 0 \le i \le q \\ R - Qq(q+1) + Q(i-q)^2 & \text{if } q < i < p \end{cases}$$

$$\text{where } q = (p-1)/2$$

yields the ith address generated by the technique for initial address R and parameter Q (approximately a quotient). This expression for f is obtained from an algebraic manipulation of the difference sequence. By making suitable restrictions on f, it is possible to obtain search techniques which are similar to certain other previously mentioned searches.

Restrictions	*Reference*
(*a*) $Q \equiv 1$ and $0 \le i < p$	Radke, 1970, Day, 1970
(*b*) $Q \equiv 1 \quad q < i \le p$	Maurer, 1968
(*c*) $1 \le Q < p \quad q < i \le p$	Bell, 1970

To show that the search method accesses the entire table, we verify the following.

Claim

If the table size p is a prime of the form $4k + 3$ for some natural number k, then the FTQQ search method generates all table addresses.

This may be seen as follows. First, observe that any particular sequence of addresses generated by the technique is equivalent mod p to the following sequence.

$$\hat{R} - q^2Q, \hat{R} - (q-1)^2Q, \ldots, \hat{R} - 9Q, \hat{R} - 4Q, \hat{R} - Q, \hat{R}, \hat{R} + Q, \hat{R} + 4Q, \ldots, \hat{R} + (q-1)^2Q, \hat{R} + q^2Q$$

where $\hat{R} = R + q^2Q$, since the sequence of differences for this sequence is identical to (1).

To substantiate the claim, it suffices to show that each of the following lemmas hold.

1. If p is a prime then there does not exist distinct integers i and j $(0 \le i, j \le q)$ such that $\hat{R} + i^2Q \equiv \hat{R} + j^2Q$.
2. If p is a prime of the form $4k + 3$ for some natural number k then there does not exist integers i and j $(0 < i \le q$ and $0 \le j \le q)$ such that $\hat{R} - i^2Q \equiv \hat{R} + j^2Q$.

Once these are shown to hold, the full table quadratic quotient search method will be seen to access the entire table since (*a*) during the first $q + 1$ accesses the addresses will all be distinct by 1, (*b*) during the last q accesses the addresses will all be distinct by 1 as well, and (*c*) there will be no duplication of addresses in the first and last set of accesses by 2. The choice of intervals for i and j in 2 is due to the way the term R is generated by the search method. This may be seen in the specification of f, where it is possible to advance an alternate (but equivalent definition) of f, i.e.,

$$f(i, Q, R) \equiv \begin{cases} R - Qi(i+1) & \text{if } 0 \le i < q \\ R - Qq(q+1) + Q(i-q)^2 & \text{if } q \le i \le p\ . \end{cases}$$

Since p is prime, the first lemma reduces to the following.

If p is a prime then there does not exist distinct integers i and j $(0 \le i, j \le q)$ such that $i^2 \equiv j^2$.

Assume not; then there exists distinct i and j as above such that $i^2 \equiv j^2$. Then $i^2 \equiv j^2$ is equivalent to $(i + j)(i - j) = np$ for some natural number n and this implies n is zero since p is not a factor of either term on the left-hand side of the equation. And we have a contradiction.

The second lemma reduces to the following.

If p is a prime of the form $4k + 3$ for some natural number k then there does not exist integers i and j $(0 < i \le q$ and $0 \le j \le q)$ such that $-i^2 \equiv j^2$.

Assume not; then there exists distinct i and j as above such that $-i^2 \equiv j^2$. Then $-i^2 \equiv j^2$ is equivalent to $i^2 + j^2 = np$ for some integer n less than p because $i \le q$ and $j \le q$. The following number theoretic result is applicable here (Dumey, 1956).

The equation $x^2 + y^2 = m$ has integer solutions for x and y if, and only if, the canonical factorisation of m into prime powers contains no factor p^e with p of the form $4k + 3$ and e odd.

Thus, if p is prime of the form $4k + 3$, there are no distinct integer solutions since $e = 1$ in this case. Note that the case $i = j = n = 0$ is relevant to the previous comment concerning the alternate definition of f.

Continuing the analysis of the FTQQ method, our attention is directed toward clustering of addresses. Let us further assume that S_p is the set of 3-permutations of Z_p, and that H_p is the set of all sequences of addresses generated by the full table quadratic quotient search. Observe that the cardinality of H_p is $p(p - 1)$. We begin by giving a characterisation of the elements of S_p.

Lemma

For every $\alpha = (a, b, c)$ in S_p either

1. $a + c - 2b \equiv 0$, or
2. there exist integers i and ξ such that

$$0 < \xi < p \text{ and}$$
$$0 \le i \le q - 2 \text{ or } q \le i \le p - 3 \text{ with}$$
$$a + c - 2b \equiv 2\xi \text{ and}$$
$$b - a \equiv 2\xi(i + 1)\ .$$

The lemma is proved by demonstrating that if $i \equiv q - 1$, $p - 2$, or $p - 1$ (when $\xi > 0$) then the element (a, b, c) cannot be in S_p. Suppose (1) does not hold, $i = q - 1$, and (a, b, c) is in S_p. Then

$$b - a \equiv 2\xi q$$
$$a + c - 2b \equiv 2\xi$$

from which it follows that $a \equiv c$, contradicting our assumption that (a, b, c) is in S_p. Similar contradictions are reached if $i = p - 2$ or $i = p - 1$ is assumed.

The next step is to show that the number of 3-permutations of Z_p produced by truncating elements of H_p is exactly $p(p - 1)(p - 2)$; that is, all 3-permutations of Z_p are produced by the search technique. Since at most $p(p - 1)(p - 2)$ 3-permutations can be produced this way because of the cardinality of H_p, it follows that every element of S_p appears as a 'subpattern' in exactly one element of H_p. Thus the secondary clustering is avoided as it is in the quadratic quotient search method.

There are three cases to consider to show an arbitrary 3-permutation of Z_p to be a 'subpattern' of an element of H_p. Let (a, b, c) be in S_p. Then the following can be verified.

1. If $a + c - 2b \equiv 0$ then let

$$i \equiv q - 1$$
$$Q \equiv b - a$$
$$R \equiv a + \xi q(q - 1)\ .$$

2. If $b - a \equiv 2\xi(j + 1), q \le j \le p - 3$, and $a + c - 2b \equiv 2\xi$ then let

$$i \equiv j$$
$$Q \equiv \xi$$
$$R \equiv a + \xi q(q + 1) - \xi(j - q)^2\ .$$

3. If $b - a \equiv 2\xi(j + 1), 0 \le j \le q - 2$, and $a + c - 2b \equiv 2\xi$ then let

$$i \equiv j$$
$$Q \equiv -\xi$$
$$R \equiv a + \xi j(j + 1)\ .$$

In all three cases

Loading factor	(A)	(B)	(C)
0·500	2·01	2·12	2·01
0·555	2·25	2·39	2·25
0·605	2·54	2·76	2·53
0·656	2·90	3·21	2·91
0·706	3·40	3·76	3·40
0·757	4·10	4·64	4·11
0·807	5·17	5·83	5·19
0·858	6·97	7·96	7·02
0·908	10·79	11·77	10·84
0·959	23·70	24·35	23·66
0·984	58·35	58·75	58·40

A ≡ full table quadratic quotient search
B ≡ full table quadratic search (Day 1970)
C ≡ full table linear quotient search (Bell and Kaman, 1970)
table size 991.

Fig. 1 Sample values of average search length

$$f(i, Q, R) \equiv a ,$$
$$f(i + 1, Q, R) \equiv b ,$$
and
$$f(i + 2, Q, R) \equiv c .$$

Thus each element of S_p can be located in a suitable element of H_p and since the cardinality of the domain of f relative to locating elements of S_p in elements of H_p is $p(p - 1)(p - 2)$ ($Q \equiv 0$ is not applicable; nor is $i = p - 1$ or $p - 2$), it follows that no element of T_p can be located somewhere other than as described above.

It follows from the above analysis that the period of the FTQQ search method is always equal to the table size and the primary and secondary clustering are avoided as well. In particular, every 3-permutation appears in exactly one sequence of addresses generated by the FTQQ search method.

4. Results

A lower bound on the average number of accesses to the table required to insert another item into the table is approximately given by

$$1/(1 - k/p) ,$$

where k is the number of items presently in the table and p is the table size. The ratio k/p is referred to as the loading factor. The simulation results, tabulated in **Fig. 1,** show the averages achieved by search technique presented here are very close to the lower bound (on the average). Results for two other full table techniques are presented as well. Each entry in the right-most three columns of the table is the average number of addresses generated to insert each of 20,000 keys into six different initial tables. The sample variance of the search length is smaller for the full table quadratic quotient search method than that for either the full table linear quotient or full table quadratic searches.

References

BELL, J. R. (1970). The quadratic quotient method: a hash code eliminating secondary clustering, *CACM*, Vol. 13, No. 2, pp. 107-109.

BELL, J. R., and KAMAN, C. H. (1970). The linear quotient hash code, *CACM*, Vol. 13, No. 11, pp. 675-677.

DAY, A. C. (1970). Full table quadratic searching for scatter storage, *CACM*, Vol. 13, No. 8, pp. 481-482.

DUMEY, A. I. (1956). Indexing for rapid random-access memory, *Computers and Automation*, Vol. 5, No. 12, pp. 6-8.

ECKER, A. (1974). The period of search for the quadratic and related hash method, *The Computer Journal*, Vol. 17, No. 4, pp. 339-342.

HOPGOOD, F. R. A., and DAVENPORT, J. (1972). The quadratic hash method when the table size is a power of 2, *The Computer Journal*, Vol. 15, No. 4, pp. 314-315.

MAURER, W. D. (1968). An improved hash code for scatter storage, *CACM*, Vol. 11, No. 1, pp. 35-38.

MORRIS, R. (1968). Scatter table techniques, *CACM*, Vol. 11, No. 1, pp. 38-44.

NIVEN, I., and ZUCKERMAN, H. S. (1966). *An Introduction to the Theory of Numbers*, Wiley and Sons, New York.

PETERSON, W. W. (1957). Addressing for random-access storage, *IBM J. Res. Dev.*, Vol. 1, pp. 130-146.

RADKE, C. E. (1970). The use of quadratic residue research, *CACM*, Vol. 13, No. 2, pp. 103-105.