Information Processing Letters 20 (1985) 179–181
North-Holland
10 May 1985

INTERPOLATION-BINARY SEARCH *

Nicola SANTORO

School of Computer Science, Carleton University, Herzberg Building, Ottawa, Ontario, Canada K1S 5B6

Jeffrey B. SIDNEY

Faculty of Administration, University of Ottawa, 550 Cumberland St., Ottawa, Ontario, Canada K1N 6N5

Communicated by W.L. Van der Poel
Received September 1984

Keywords: Ordered array, searching, multiple complexity objective

1. Introduction

Consider an array of n records stored so that the keys are arranged in increasing order. Using binary search, the search for a record can be performed in time O(log n), both in the average and in the worst case. If the keys are uniformly distributed, an improved expected performance can be obtained by interpolation [2–6] or, for $n < 500$, by interpolation-sequential [1] search. Namely, the search can be performed in expected time O(log log n) by interpolation search, and $(n\pi/32)^{1/2} + O(1)$ by interpolation-sequential search. Unfortunately, both techniques require O(n) time in the worst case.

In this paper it is shown that it is possible to search in O(log n) time in the worst case and, if the keys are uniformly distributed, in O(log log n) time in the average. This result is achieved by interpolation-binary search, a new technique which combines 'interpolation' and 'binary' steps; thus, the proposed algorithm represents an improvement on the existing methods.

* This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants A2415 and A2507.

2. The algorithm

Let $X = \langle X_1, \ldots, X_n \rangle$ be an ordered vector of elements in the interval (0, 1), where $X_i \leqslant X_{i+1}$ $(1 \leqslant i \leqslant n-1)$. Given $\alpha \in (0, 1)$, the algorithm below will either find an index i such that $X_i = \alpha$, or determine that no such i exists. Note that the algorithm has two parameters θ and S which are fixed by the user as discussed later.

Interpolation-Binary Search. (*Input*: α, θ, S.)

(1) $L := 0$, $H := n$, $X_L := 0$, $y := \alpha$, $N := n$.

(2) **If** $N < S$ **go to** step (4). Otherwise, set $\hat{L} := \mathrm{Max}\{L + \lceil Ny - \theta(Ny(1-y))^{1/2} \rceil, L+1\}$, $\hat{H} := \mathrm{Min}\{L + \lfloor Ny - \theta(Ny(1-y))^{1/2} \rfloor, H-1\}$.

(3) **If** $X_{\hat{L}} = \alpha$ or $X_{\hat{H}} = \alpha$ **then** stop. Otherwise

  (a) **if** $\alpha < X_{\hat{L}}$ set $H := \hat{L}$; **else**
  if $X_{\hat{L}} < \alpha < X_{\hat{H}}$ set $L := \hat{L}$, $H := \hat{H}$; **else**
  if $X_{\hat{H}} < \alpha$ set $L := \hat{H}$;

  (b) $y := (\alpha - X_L)/(X_H - X_L)$.

(4) $M := \lceil \frac{1}{2}(L+H) \rceil$.

(5) **If** $X_M = \alpha$ **then** stop. Otherwise

  (a) **If** $\alpha < X_M$ set $H := M$, **else** set $L := M$;

  (b) $N := H - L - 1$,
  $y := (\alpha - X_L)/(X_H - X_L)$;

  (c) **if** $N = 0$ stop (α is not in X). Otherwise **go to** step (2).

Note that steps (2) and (3) correspond to one step of interpolation search, while steps (4) and (5) constitute a step of binary search.

In the discussion below, all logarithms are base 2, and an iteration is said to begin with each execution of step (2). Also, the 'size' of a problem is defined to be $H - L - 1$.

Theorem 2.1. *The Interpolation-Binary Search performs at most* $\lfloor \log n \rfloor$ *iterations.*

Proof. Let N_k be the number of elements remaining after k executions of step (5). From steps (4) and (5) it is obvious that $N_k \leqslant N/2^k$. The result follows from the inequality $N_k > 1$. □

In order to establish the average case performance of Interpolation-Binary Search, the following uniformity assumption is made.

Uniformity Assumption. *For* $1 \leqslant i \leqslant n$, X_i *is the* i*th smallest element (order statistic) from a sample of* n *independent trials obtained from the uniform distribution on the interval* $(0, 1)$.

Lemma 2.2. *Let* $\theta > 1$ *and* $N \geqslant \frac{1}{2}\theta^2$. *With probability* $p \geqslant 1 - (1/\theta^2)$ *one iteration reduces a problem of size* N *to a problem of size less than*

$$\theta(Ny(1-y))^{1/2}.$$

Proof. Call an iteration a 'success' if either termination occurs or the second condition of step (3)(a) holds. It suffices to show that the probability of a success at each iteration is greater than or equal to $1 - (1/\theta^2)$, since the interval resulting from a nonterminating success will be halved by the binary search steps (4) and (5). Let $Z_N(a, b)$ be a random variable defined to be the number of elements not greater than α in a sample of N independent trials drawn from the uniform distribution on (a, b).

$Z_N(a, b)$ has a binomial distribution with parameters N and $y = (\alpha - b)/(b - a)$, with mean $\mu = Ny$ and variance $\sigma^2 = Ny(1 - y)$. By Chebyshev's inequality,

$$P\left[|Z_N(a, b) - yN| < Q\sigma\right] \geqslant 1 - (\sigma^2/\theta^2\sigma^2)$$

for $\theta > 1$, or

$$P\left[|Z_N(a, b) - yN| < \theta(Ny(1-y))^{1/2}\right] \geqslant 1 - (1/\theta^2).$$

The lemma follows since the conditional distribution of $\{X_{L+1}, \ldots, X_{H-1}\}$ given X_L and X_H is that of the order statistics of a sample of $H - L - 1$ independent trials drawn from the uniform distribution on $[X_L, X_H]$. □

Lemma 2.3. *Let* $\theta > 1$ *and* $S = \frac{1}{2}\theta^2$. *The expected number of iterations of Interpolation-Binary Search until* $N < S$ *is no more than* $(1/p)\lceil \log \log n \rceil$, *where* $p = 1 - (1/\theta^2)$.

Proof. Whenever $N \geqslant S = \frac{1}{2}\theta^2$, Lemma 2.2 implies that a successful iteration will reduce a problem of size N to one of size less than $2\theta(Ny(1 - y))^{1/2}$ at the end of step (3). Steps (4) and (5) guarantee a further reduction to size less than

$$\theta(Ny(1-y))^{1/2} \leqslant \theta\left(N\left(\tfrac{1}{2}\right)^2\right)^{1/2} = \tfrac{1}{2}\theta N^{1/2}.$$

A simple induction shows that after k iterations a problem of size n will be reduced to one of size less than $\frac{1}{4}(\theta^2 n^{(1/2)^k})$. Solving the inequality

$$\tfrac{1}{4}(\theta^2 n^{(1/2)^k}) \geqslant S = \tfrac{1}{2}\theta^2$$

yields the result $k \leqslant \log \log n$. Thus, $\lfloor \log \log n + 1 \rfloor$ successes are guaranteed to reduce the size from n to $N < S$. The probability of a success at any iteration is not smaller than $p = 1 - (1/\theta^2)$ by Lemma 2.2. By comparison with the geometric distribution with probability p of success, the expected number of iterations until a success is at most $1/p$. The lemma now immediately follows. □

Theorem 2.4. *The expected number of iterations of Interpolation-Binary Search is at most* $\frac{4}{3}\lceil \log \log n + 2 \rceil$.

Proof. Choose $\theta = 2$ and $S = \frac{1}{2}\theta^2 = 2$. Then, by Lemma 2.3, the problem will be reduced to size $N < S = 2$ in at most $\frac{4}{3}\lceil \log \log n + 1 \rceil$ iterations. One additional iteration guarantees a solution. □

References

[1] G.H. Gonnet and L.D. Rogers, The interpolation-sequential search algorithm, Inform. Process. Lett. 6 (4) (1977) 136–139.

[2] G.H. Gonnet, L.D. Rogers and A. George, An algorithmic and complexity analysis of interpolation search, Acta Informatic 13 (1) (1980) 39–46.

[3] M. van der Nat, On interpolation search, Comm. ACM 22 (12) (1979) 681.

[4] Y. Perl and E.M. Reingold, Understanding the complexity of interpolation search, Inform. Process. Lett. 6 (6) (1977) 219–221.

[5] Y. Perl and H. Avni, Interpolation search—a log log N search, Comm. ACM 21 (7) (1978) 550–553.

[6] A.C. Yao and F.F. Yao, The complexity of searching an ordered random table, Proc. 17th FOCS, Houston (1976) 173–176.