ON WORST CASE ROBIN-HOOD HASHING

Luc Devroye, Pat Morin
School of Computer Science
McGill University
Montreal, Canada H3A 2K6
luc@cs.mcgill.ca

and

Alfredo Viola
Pedeciba Informatica
Districto 6, Casilla de Correo 16120
Universidad de la República
Montevideo
Uruguay

September 24, 2003

Abstract. We consider open addressing hashing, and implement it by using the Robin Hood strategy, that is, in case of collision, the element that has traveled the furthest can stay in the slot. We hash $\sim \alpha n$ elements into a table of size n where each probe is independent and uniformly distributed over the table, and $\alpha < 1$ is a constant. Let M_n be the maximum search time for any of the elements in the table. We show that with probability tending to one, $M_n \in [\log_2 \log n + \sigma, \log_2 \log n + \tau]$ for some constants σ, τ depending upon α only. This is an exponential improvement over the maximum search time in case of the standard FCFS (first come first served) collision strategy, and virtually matches the performance of multiple choice hash methods.

Keywords and phrases. Open addressing, hashing, Robin Hood, worst-case search time, collision resolution, probabilistic ananlysis of algorithms.

CR Categories: 3.74, 5.25, 5.5.

1991 Mathematics Subject Classifications: 60D05, 68U05.

The first two authors' research was supported by NSERC Grant A3456 and FCAR Grant 90-ER-0291. The third author was supported by Proyectos de investigación CSIC fondos 2000-2002 and 2002-2004 at Universidad de la República.

Introduction

In hashing with chaining with a table of size n holding $m = \lceil \alpha n \rceil$ elements, where $\alpha > 0$ is a constant, the worst-case search time is equal to the length of the longest chain. If the hash values are independent and uniformly distributed over the table, then the maximum chain length is asymptotic to $\log n / \log \log n$ in probability (Gonnet, 1981; Devroye, 1985), for any fixed value of α.

In this paper we consider open addressing hashing with random probing. A table of size n is given, into which we place $m = \lceil \alpha n \rceil$ elements, where $\alpha \in (0, 1)$ is a fixed constant. Each element has associated with it an infinite probe sequence consisting of i.i.d. integers uniformly distributed over $\{1, \ldots, n\}$, representing the consecutive places of probes for that element. It is assumed that when searching for an element, its infinite probe sequence is available to the searcher. The probe sequence for the i-th element is denoted by $X_{i,0}, X_{i,1}, X_{i,2}, \ldots$. Elements are inserted sequentially into the table. If the i-th element is placed in position $X_{i,j}$, then we say that the i-th element has age j, as it requires j hops to reach the element in case of a search. When the i-th element of age j and the i'-th element of age j' compete for the same slot ($X_{i,j} = X_{i',j'}$), a collision resolution strategy is needed. Several collision resolution strategies have dominated the literature.

The standard open addressing method resolves the collision by giving the place to the first key to arrive there according to a first come first served policy (FCFS), so the test is based on $\min(i, i')$. Amble and Knuth (1974) suggested the idea that *any* of the colliding elements could get the position in the hope of speeding up unsuccesful searches. Note that for random probing, for any strategy that does not look ahead, the sum of the ages of all elements in a hash table has a distribution that is independent of the collision resolution strategy. There are differences though when one considers the maximal age among all elements in a table. Two of the strategies that do not look ahead before deciding which element should get the position are the LCFS (last come first served) heuristic (Poblete and Munro, 1989), in which the position is given to the last element that arrives (thus, using $\max(i, i')$), and the Robin Hood strategy (Celis, 1986; Celis, Larson and Munro, 1985; Viola and Poblete, 1998), in which the position is given to the element that is furthest away from its home location (the element corresponding to $\max(j, j')$). The Robin Hood strategy tends to equalize the ages of all inserted elements (hence the name Robin Hood), thus reducing the maximum successful search time. Both FCFS and Robin Hood decrease the variance of the search time. As pointed out earlier, for random probing, the expected search time for a single random element is identical for all collision resolution strategies that do not look ahead. An interesting property of Robin Hood is that every permutation of the insertion sequence produces the same final hash table, provided that a consistent tie breaker is used (for example, $\min(i, i')$).

In open addressing hashing, most of the proposed schemes to improve the search cost of a random element in a hash table (like Brent's method, binary tree, optimal and min-max hashing) have very high cost for table creation. Other methods like multiple choice hashing are more inefficient in the use of space. As presented in Celis (1986), Robin Hood is an open addressing hashing scheme that is as simple to program as the standard algorithm, takes only $\Theta(n \log n)$ on the average to load a full table, requires no additional memory for insertions and has very small variance. This last fact is a key observation in Celis (1986), to speed up the searching cost of a random element. The main idea is not to probe the first position in the probe sequence, but on the most probable place and then move away from it in both directions.

For uniform probing (that is, a probe sequence without repetition) the expected value of the longest probe sequence for the standard FCFS algorithm for α-full tables ($\alpha < 1$) is $\log_{1/\alpha} n - \log_{1/\alpha}(\log_{1/\alpha} n) + O(1)$ and for full tables is $0.631587\ldots \times n + O(1)$ (Gonnet, 1981).

Poblete and Munro (1989) prove that for random probing (that is, a probe sequence with repetition) the expected value of the longest probe sequence for the LCFS heuristic is bounded by

$$1 + \Gamma^{-1}(\alpha n)\left(1 + \frac{\log\log(1/(1-\alpha))}{\log \Gamma^{-1}(\alpha n)} + O\left(\frac{1}{\log^2 \Gamma^{-1}(\alpha n)}\right)\right),$$

where Γ is the Gamma function, and

$$\Gamma^{-1}(\alpha n) = \frac{\log n}{\log\log n}\left(1 + \frac{\log\log\log n}{\log\log n} + O\left(\frac{1}{\log\log n}\right)\right).$$

Although this is not a tight bound, this was the first open addressing method for which a sub-logarithmic bound in n was proven.

Celis (1986) proves that the expected value of the longest probe seqence for random probing and a full Robin Hood hash table ($\alpha = 1$) is $\Theta(\log n)$. Moreover, when $\alpha < 1$ he proved that for random probing, the expected value of the longest probe sequence for the Robin Hood heuristic is bounded by $3(H_n - H_{n-m})/\alpha + \lceil \log(n-2) \rceil$, where $H_n = \sum_{1 \le i \le n} 1/i$. This bound is improved in this paper to $\log_2 \log n$. For further discussions and results, see Knuth (1998), Vitter and Flajolet (1990), Gonnet and Baeza-Yates (1991) or Flajolet, Poblete and Viola (1998). It is perhaps worth reproducing Table 5.9 from Celis's dissertation, in which empirical estimates were computed for the longest successful probe length with the Robin Hood strategy, which suggests a $\Theta(\log\log n)$ complexity for the problem when $\alpha < 1$.

n	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 0.9$	$\alpha = 1.0$
1021	$3.629 \pm .065$	$4.000 \pm .013$	$4.329 \pm .064$	$5.105 \pm .041$	$10.443 \pm .187$
4093	$3.967 \pm .024$	$4.062 \pm .033$	$4.800 \pm .054$	$5.329 \pm .064$	$12.133 \pm .208$
16273	$4.014 \pm .016$	$4.262 \pm .060$	$5.000 \pm .000$	$5.771 \pm .057$	$13.819 \pm .172$
65537	$4.029 \pm .023$	$4.614 \pm .066$	$5.000 \pm .000$	$6.000 \pm .000$	$15.181 \pm .178$
262139	$4.098 \pm .040$	$4.967 \pm .024$	$5.022 \pm .020$	$6.000 \pm .000$	$16.815 \pm .179$

Expected length of longest successful probe sequence.

It is perhaps worthy of mention that there are several other ways other of obtaining dynamic hash tables with $O(\log \log n)$ expected maximum successful search times. Consider hashing with chaining, and let the elements have a choice of two randomly picked positions. An element is placed into the slot with the least number of elements (at the time of insertion). This simple double choice shows that the maximum slot occupancy is in probability asymptotic to $\log_2 \log_2 n$ (Azar, Broder, Karlin and Upfal, 1999; Broder and Karlin, 1990; Czumaj and Stemann, 1997; Mitzenmacher, 1997).

There has been interest in obtaining $O(1)$ expected worst-case performance, or even $O(1)$ deterministic worst-case performance for search in hash tables. For static hash tables, Fredman, Komlós and Szemerédi (1984) proposed a solution. Czumaj and Stemann (1997) showed that if each element has two randomly chosen hash positions, then with high probability, a static (off-line) chaining hash table can be constructed that has worst chain length 2, provided that the table size is at least αn for some threshold constant α. For dynamic hash tables, the early research was in the direction of dynamic perfect hash functions (Dietzfelbinger and Meyer auf der Heide (1990), Dietzfelbinger, Gil, Matias and Pippenger (1992), Dietzfelbinger, Karlin, Mehlhorn, Meyer auf der Heide, Rohnert and Tarjan (1994), Brodnik and Munro (1999)). Cuckoo hashing (Pagh and Rodler, 2001) is also an attempt in this direction. It stands out though through its simplicity and the promising experimental results reported by Pagh and Rodler: each of m data points has two hash functions, one to be used in each of two tables of size $n \geq (1 + \epsilon)m$. The element must be placed in one of the tables at one of the two locations. Upon insertion of a new element, old elements get kicked out and move around, kicking out other elements if necessary, until either a loop is detected or the insertion process halts. In case of a loop, the entire table is rehashed. The expected time for an insertion is still $O(1)$, and the worst-case successful search time is bounded by 2. However, one needs a powerful collection of hash functions, as each rehash operation requires an entirely new and independent set of hash values.

Let us denote by M_n the maximal successful search time, that is, the maximal age among any of the m elements in the hash table, and by T_n the maximum insertion cost of an element. In a FCFS strategy, we note that $M_n = T_n - 1$, but this is no longer true with other strategies. In fact, in this paper we show the following.

THEOREM 1. *In open addressing with Robin Hood collision resolution, there exists a constant C depending upon α only such that*

$$\lim_{n\to\infty} \mathrm{P}\{M_n \geq \log_2 \log n + C\} = 0 \ .$$

We will see that $C \to \infty$ when $\alpha \to 1$, so this result is meaningful only when $\alpha < 1$. The result above implies an exponential improvement over the FCFS strategy. Furthermore, this bound is optimal modulo a finite constant:

THEOREM 2. *In open addressing with Robin Hood collision resolution (and any method of breaking ties),*

$$\mathrm{P}\{M_n \leq \log_2 \log n - \log_2(6\log(8/\alpha))\} = O(1/\sqrt{n}) \ .$$

The implications of this are not be underestimated, as open addressing tables are the oldest and simplest hashing structures. The multiple choice hashing methods in their original form are intrinsically chaining methods, and thus slightly more inefficient spacewise.

The $\log_2 \log n$ behavior follows, roughly speaking, from the following observation. If we place all m elements in their first choice bins, then all but one element from each bin must move to another bin. The number of these excess elements is about m^2/n times a constant. Just looking at these displaced elements, we repeat the argument k times, obtaining increasingly smallest sets to be displaced. After k steps, the number of elements left is of the order of $n(m/n)^{2^k}$, or $n\alpha^{2^k}$. This is of constant order when k is about $\log_2 \log n$.

Balls in urns

Throw m balls uniformly at random into n urns. Let urn i receive N_i balls, and define

$$A = \sum_{i=1}^{n} (N_i - 1)_+$$

the number of balls left after removing one ball from each occupied urn. We say that A has the (m, n) urn distribution.

THE (m, n) URN DISTRIBUTION. Let A have the (m, n) urn distribution. Then

$$\mathrm{E}\{A\} = \sum_{j=1}^{m} \left(1 - (1 - 1/n)^{j-1}\right) \ .$$

Note that $(1 - 1/n)^m \geq 1 - m/n$ and $(1 - 1/n)^m \leq 1 - m/n + m(m-1)/2n^2$, so that

$$
\begin{aligned}
\frac{m^2}{2n} \geq \frac{m(m-1)}{2n} \\
&= \sum_{j=1}^{m-1} \frac{j}{n} \\
&\geq \mathbf{E}\{A\} \\
&\geq \sum_{j=1}^{m-1} \frac{j}{n} - \sum_{j=1}^{m-1} \frac{j(j-1)}{2n^2} \qquad (1) \\
&= \frac{m(m-1)}{2n} - \frac{m(m-1)(m-2)}{6n^2} \\
&\geq \frac{m(m-1)}{3n} \\
&\geq \frac{m^2}{4n} (\text{the last step is true only if } m \geq 4) \ .
\end{aligned}
$$

We also need some concentration inequalities for A. To present these inequalities, let $(X_1, \ldots, X_n)$ be a vector of independent random variables (on an arbitrary measurable space S), let $f : S \to \mathbf{R}$ be a measurable function, and set

$$Z = f(X_1, \ldots, X_m) \ .$$

Let $X'_1, \ldots, X'_m$ be independent copies of $X_1, \ldots, X_m$, and write

$$Z^{(i)} = f(X_1, \ldots, X_{i-1}, X'_i, X_{i+1}, \ldots, X_m).$$

The Efron-Stein inequality (Efron and Stein, 1981; Steele, 1986) states that

$$\mathbf{V}\{Z\} \leq \frac{1}{2} \mathbf{E} \left\{ \sum_{i=1}^{m} (Z - Z^{(i)})^2 \right\} \ .$$

If $Z \equiv A$, and $X_1, \ldots, X_m$ are the urns chosen by elements 1 through m, and X'_i is independent of the X_j's and distributed as X_i, then $|Z^{(i)} - Z| \leq 1$. Thus, $\mathbf{V}\{Z\} \leq m/2$. With the inequalities for $\mathbf{E}\{A\}$ taken into account, we have, by Chebyshev's inequality, for all $t > 0$,

$$\mathbf{P}\{|A - \mathbf{E}\{A\}| \geq t\} \leq \frac{\mathbf{V}\{A\}}{t^2} \leq \frac{m}{2t^2} \ .$$

The head-and-belly view

The construction of the hash table may be looked at in a global manner for Robin Hood strategies, since every permutation of the input sequence produces the same hash table. We start by placing all elements at their first choices $X_{i,0}$, $1 \leq i \leq m$. Some bins in the table may have many elements, but that is acceptable. We call this the first stage. At the k-th stage in our construction, picture a hash table ("the head") containing elements of age k, possibly many per cell, and a second hash table ("the belly") containing at most one element per cell, and that element is of age less than k. Furthermore—and this is crucial—,

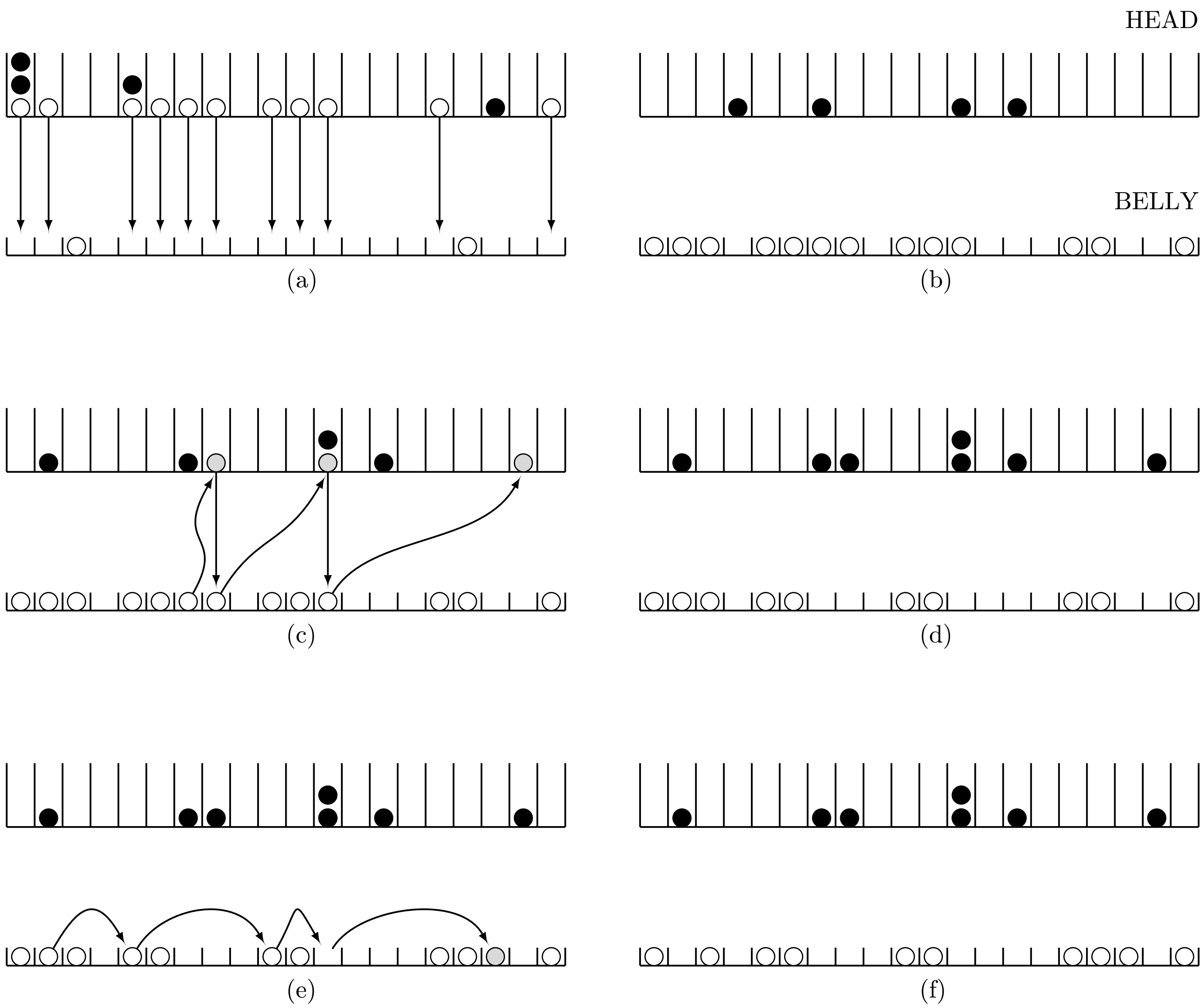

FIGURE 1. In (a), we show a $(k-1)$-head that is not empty. The elements in position one in their bins (in white) move to the belly (step B). The other elements (in black) move to a random position in the k-head, shown in figure (b). This is step A. Clearly, there are some conflicts between head and belly in (b). In step C, these are resolved. For each conflict, an element in the belly is taken, and is moved to a random position in the belly. For example, in (c), we show the moves of an element, as it first ages to age k (so that its randomly picked position lands it in the head), which triggers a new conflict in the belly, which is immediately taken care of by letting that element move to a random position, which again happens to be in the head (gray element), and finally, the last conflict generated leads to yet another element in the head, causing no further conflicts. The resulting configuration is (d). In (e), the last remaining conflict is taken care of by random hops, resulting in the final configuration (f) of the k-belly and k-head. In example (e), all hops remain in the belly, and result finally in a cell in the belly being filled with a new element.

if cell i in the head is occupied, then cell i is empty in the belly. This head-and-belly view allows us to proceed, by letting k grow until finally the head is empty, and all elements are in the belly.

The belly is initially empty, and all elements are in the head, in stage one. Given the $(k-1)$-st stage situation, we construct the k-th stage as follows.

A. All elements in the $(k-1)$-stage head that are in positions two and above in their bins move to a randomly selected bin in the k-head.

B. The remaining elements of the $(k-1)$-head (at most one per cell) are added to the $(k-1)$-belly (in the corresponding position). Note that this may create some conflicts with the k-head just created.

C. While there is a head-belly conflict, take a conflicting element in the belly (that is, an element in cell i, such that the k-head also has an element in cell i), and let it start hopping uniformly and ranomly (and aging by oner with each hop), according to the rules of Robin Hood hashing, until it, or the element it causes to move, finds a position in a cell by itself in the belly, without conflict with the k-head, or a position in the k-head (an element that reaches age k must move to the k-head). In the latter case, a new conflict may be triggered. At the end of this, there is no further conflict, and the resulting tables are called the k-head and the k-belly.

LEMMA 1. *Let N be the number of elements added to the k-head in step C, given that the k-head has at most K elements to start with after steps A and B, and given any distribution of elements in belly and head at that point. Then, with $\lambda = 1/\log(1/\alpha)$, N is stochastically smaller than $\lambda G_K + K$, where G_K is a gamma (K) random variable. In particular,* $\mathrm{E}\{N\} \leq (\lambda+1)K$, *and*

$$\mathrm{P}\{N \geq (2\lambda+1)K\} \leq \left(\frac{2}{e}\right)^K .$$

PROOF. There are initially at most K elements in the belly that can cause a conflict with the head. When these elements move, at each step we have a probability at least $1-\alpha$ of finding an empty slot (empty for both head and belly). When such a slot is found, the chain of moves ends. In each step, at most one element moves to the k-head. The number of additions to the k-head to just eliminate one belly conflict is thus stochastically smaller than one plus a geometric (α) random variable Y:

$$\mathrm{P}\{Y \geq i\} \leq \alpha^i \ , \ i \geq 0 \ .$$

If E is unit exponential, then we see that

$$\mathrm{P}\{\lambda E \geq i\} = \exp\left(-\frac{i}{\lambda}\right) = \alpha^i$$

provided that $\alpha = \exp(-1/\lambda)$, or $\lambda = 1/\log(1/\alpha)$. Therefore, $Y \prec E/\log(1/\alpha)$. Since we have to eliminate K possible conflict elements in the belly, the total number of elements added to the head is stochastically smaller than $K + Y_1 + \cdots Y_K$, where the Y_i's are independent and are all stochastically dominated by λE. Thus, if $E_1, \ldots, E_K$ are independent exponential random variables, and G_K is a gamma (K) random variable, we see that the number N of additions to the head in part C is stochastically smaller than $K + \lambda(E_1 + \cdots + E_K) \stackrel{\mathcal{L}}{=} K + \lambda G_K$. In other words,

$$\begin{aligned}
\mathrm{P}\{N \geq (2\lambda + 1)K\} &\leq \mathrm{P}\{\lambda G_K \geq 2\lambda K\} \\
&= \mathrm{P}\{G_K \geq 2K\} \\
&\leq \mathrm{E}\left\{e^{tG_K} e^{-2tK}\right\} \quad \text{(any } t > 0\text{)} \\
&= \left(\frac{e^{-2t}}{1-t}\right)^K \\
&= \left(\frac{2}{e}\right)^K \quad \text{(take } t = 1/2\text{)}
\end{aligned}$$

This concludes the proof of Lemma 1. □

It is important to note that if $\alpha \to 1$, then $\lambda \to \infty$, so the results below are meaningful only when $\alpha < 1$.

Lemma 2. *Define $b = (2\lambda + 2)\alpha$ and assume that $b < 1$. Let D be the integer*

$$D = \left\lfloor \log_2\left(\frac{2}{3\log(1/b)}\right) - 0.1 \right\rfloor .$$

Let Z be the number of elements in the r-head, with $r = \lfloor \log_2 \log n \rfloor + D$. Then

$$\lim_{n\to\infty} \mathrm{P}\left\{Z \geq \frac{nb^{2^r}}{2\lambda + 2}\right\} = 0 .$$

In particular,

$$\lim_{n\to\infty} \mathrm{P}\left\{Z \geq n^{1-1/(6\times 2^{0.1})}\right\} = 0 .$$

Proof. Given that the $k-1$-head has K elements or less, then if A denotes the number of elements in the k-head after step A (not including steps B and C), we have

$$\mathrm{E}\{A\} \leq \frac{K^2}{2n}$$

and

$$\mathbb{P}\{|A - \mathbb{E}\{A\}| \geq t\} \leq \frac{K}{2t^2} \ .$$

In particular, we note that

$$\mathbb{P}\left\{A \geq \frac{K^2}{n}\right\} \leq \frac{2n^2}{K^3} \ .$$

After steps B and C, N more elements are added to the k-head. We have

$$\begin{aligned}
\mathbb{P}\left\{A + N \geq \frac{(2\lambda+2)K^2}{n}\right\} &\leq \mathbb{P}\left\{A \geq \frac{K^2}{n}\right\} + \mathbb{P}\left\{N \geq \frac{(2\lambda+1)K^2}{n} \Big| A \leq \frac{K^2}{n}\right\} \\
&\leq \frac{2n^2}{K^3} + \left(\frac{2}{e}\right)^{\frac{K^2}{n}} \ .
\end{aligned}$$

Now define the sequence a_k by $a_0 = m$,

$$a_{k+1} = \frac{(2\lambda+2)a_k^2}{n} \ .$$

Then it is easy to see that for $k > 0$,

$$a_k = \frac{n}{2\lambda+2}\left(\frac{(2\lambda+2)a_0}{n}\right)^{2^k} = \frac{n}{2\lambda+2}\left((2\lambda+2)\alpha\right)^{2^k} \ .$$

Let A_k, N_k denote the k-head cardinalities as defined above. Then

$$\begin{aligned}
\mathbb{P}\{A_r + N_r \geq a_r\} \leq\ & \mathbb{P}\{A_r + N_r \geq a_r \mid A_{r-1} + N_{r-1} \leq a_{r-1}\} \\
&+ \mathbb{P}\{A_{r-1} + N_{r-1} \geq a_{r-1} \mid A_{r-2} + N_{r-2} \leq a_{r-2}\} \\
&+ \cdots + \mathbb{P}\{A_1 + N_1 \geq a_1 \mid A_0 + N_0 \leq a_0\} \ ,
\end{aligned}$$

since $A_0 + N_0 = m = a_0$. By the definition of the a_k sequence, we note that the general term

$$\mathbb{P}\{A_k + N_k \geq a_k \mid A_{k-1} + N_{k-1} \leq a_{k-1}\}$$

is bounded by

$$\frac{2n^2}{a_{k-1}^3} + \left(\frac{2}{e}\right)^{\frac{a_{k-1}^2}{n}} \ .$$

Thus, defining $b = (2\lambda+2)\alpha$, and assuming that $b < 1$, we have

$$\begin{aligned}
\mathbb{P}\{A_r + N_r \geq a_r\} &\leq \sum_{k=0}^{r-1}\left(\frac{2n^2}{a_k^3} + \left(\frac{2}{e}\right)^{\frac{a_k^2}{n}}\right) \\
&= \sum_{k=0}^{r-1}\left(\frac{2(2\lambda+2)^3}{n\left((2\lambda+2)\alpha\right)^{3\times 2^k}} + \left(\frac{2}{e}\right)^{\frac{((2\lambda+2)\alpha)^{2^{k+1}}n}{(2\lambda+2)^2}}\right) \\
&\leq C\left(\frac{1}{nb^{3\times 2^{r-1}}} + \left(\frac{2}{e}\right)^{\frac{nb^{2^r}}{(2\lambda+2)^2}}\right) \\
&\qquad \text{(for some constant } C) \ .
\end{aligned}$$

Let $r = \lfloor \log_2 \log n \rfloor + D$ for some integer D. Then $2^{D-1} \log n \leq 2^r \leq 2^D \log n$, and $nb^{2^r} \geq nb^{3 \times 2^{r-1}} \geq nb^{(3/2)2^D \log n} = n^{1+(3/2)2^D \log b}$. Thus, if $2^D \log(1/b) < 2/3$, then

$$\lim_{n\to\infty} \mathbf{P}\{A_r + N_r \geq a_r\} = 0 \ .$$

The last statement follows from the fact that

$$\begin{aligned} nb^{2^r} &\leq nb^{2^{D-1}\log n} \\ &= n^{1-2^{D-1}\log(1/b)} \\ &\leq n^{1-2^{\log_2\left(\frac{2}{3\log(1/b)}\right)-2.1}\log(1/b)} \\ &= n^{1-1/(6\times 2^{0.1})} \ . \end{aligned}$$

This concludes the proof of Lemma 2. $\square$

REMARK. The condition $b = (2\lambda + 2)\alpha < 1$ reduces to $(2 + 2/\log(1/\alpha))\alpha < 1$. This is satisfied if $\alpha \leq 0.306891\ldots$.

LEMMA 3. *Let r be as in Lemma 2. Then the probability that the $(r+3)$-head has at least one element is $o(1)$. Thus, with probability tending to one, the maximum successful search time is bounded by $r+2$.*

PROOF. Let r be as in Lemma 2, and let Z be the number of elements in the r-head. Then it is of interest to study Z_j, the number of elements in the $(r+j)$-head for $j > 0$. Recall that $a_r \leq n^{1-1/(6\times 2^{0.1})}$. Given Z, we have $\mathbf{E}\{Z_1 \mid Z\} \leq (2+\lambda)Z^2/2n$, where we used (1) and Lemma 1. On $Z \leq a_r$, we have $\mathbf{E}\{Z_1 \mid Z\} \leq (2+\lambda)a_r^2/2n \leq (2+\lambda)n^{1-2^{0.9}/6}$. Thus, $\mathbf{P}\left\{Z_1 > (2+\lambda)\log n \times n^{1-2^{0.9}/6} \mid Z\right\} \leq 1/\log n$ by Markov's inequality, on $Z \leq a_r$. Next, on $Z_1 \leq (2+\lambda)\log n \times n^{1-2^{0.9}/6}$,

$$\mathbf{E}\{Z_2 \mid Z_1\} \leq (2+\lambda)\left((2+\lambda)\log n \times n^{1-2^{0.9}/6}\right)^2 /2n < (2+\lambda)^3 \log^2 n \times n^{1-2^{1.9}/6} \ .$$

Thus,

$$\mathbf{P}\left\{Z_2 > (2+\lambda)^3 \log^3 n \times n^{1-2^{1.9}/6} \mid Z_1\right\} \leq \frac{1}{\log n} \ .$$

Finally, on $Z_2 \leq (2+\lambda)^3 \log^3 n \times n^{1-2^{1.9}/6}$,

$$\mathbf{E}\{Z_3 \mid Z_2\} \leq (2+\lambda)\left((2+\lambda)^3\log^3 n \times n^{1-2^{1.9}/6}\right)^2 /2n < (2+\lambda)^7 \log^6 n \times n^{1-2^{2.9}/6} = o(1) \ .$$

Thus,

$$\mathbf{P}\{Z_3 \geq 1 \mid Z_2\} \leq \mathbf{E}\{Z_3 \mid Z_2\} = o(1) \ .$$

Thus,

$$
\begin{aligned}
\mathrm{P}\{Z_3 > 0\} \leq \mathrm{P}&\left\{Z \geq n^{1-1/(6\times 2^{0.1})}\right\} \\
&+\mathrm{P}\left\{Z_1 \geq (2+\lambda)\log n \times n^{1-2^{0.9}/6} \mid Z \leq n^{1-1/6\times 2^{0.1}}\right\} \\
&+\mathrm{P}\left\{Z_2 \geq (2+\lambda)^3 \log^3 n \times n^{1-2^{1.9}/6} \mid Z_1 \leq (2+\lambda)\log n \times n^{1-2^{0.9}/6}\right\} \\
&+\mathrm{P}\left\{Z_3 \geq 1 \mid Z_2 \leq (2+\lambda)^3 \log^3 n \times n^{1-2^{1.9}/6}\right\} \\
= o(1).\ \square
\end{aligned}
$$

Thus far, we have shown that if $\alpha \leq 0..306891\ldots$, the probability that the maximal displacement of any element is more than $\log_2 \log n + C$ for a constant C depending upon α only tends to zero. This matches the lower bound that we will present further on. We will now fill the gap and show this result for all α.

Proof of Theorem 1

In the proof, we let $m = \lfloor \alpha n \rfloor$ without loss of generality. We define the level of an element as the number of probes required to locate it. The level is one if the element is stored at its original location. (thus, the level is one more than the age of an element). We call the level of a cell in the table the level of the element occupying the cell if the cell is occupied, and zero otherwise. At time t, when the table holds t elements, we define

$$N_t(i) = \#\ \text{elements of level} \geq i\ .$$

Note that $N_t(i)$ is monotone in t for fixed i. When inserting the t-th element, let K_t be the number of cells probed. Clearly, K_t is geometric:

$$\mathrm{P}\{K_t = k\} = \left(1 - \frac{t-1}{n}\right)\left(\frac{t-1}{n}\right)^{k-1}\ ,\ k \geq 1.$$

We begin with a rough tail bound for $N_t(i)$.

Lemma 4. *Define*

$$\beta = \frac{2(1+\alpha)}{(1-\alpha)\log((1+\alpha)/2\alpha)}.$$

Then for all $t \leq m$, $i \geq 1$,

$$\mathrm{P}\left\{N_t(i) \geq \beta m \alpha^{i-1}\right\} \leq \mathrm{P}\left\{N_m(i) \geq \beta m \alpha^{i-1}\right\} \leq \exp\left(-\frac{1+\alpha}{1-\alpha} m \alpha^{i-1}\right)\ .$$

PROOF. When we insert the t-th element, we can increase the number of elements of level $\geq i$ by at most $(K_t - i)_+$. Therefore,

$$N_t(i) \leq \sum_{j=1}^{t} (K_j - i)_+ \ ,$$

where $K_1, K_2, \ldots, K_t$ are independent. As $K_1 \prec K_2 \prec \cdots \prec K_t$ (where $\prec$ denotes stochastic ordering), we see that

$$N_t(i) \prec \sum_{j=1}^{t} (K_{t,j} - i)_+$$

where $K_{t,1}, \ldots, K_{t,t}$ are i.i.d. and distributed as K_t. We will use Chernoff bounding (Chernoff, 1952; Hoeffding, 1963). Let $\lambda, u > 0$. Then

$$\begin{aligned}
\mathbf{P}\{N_t(i) \geq u\} &\leq \mathbf{P}\{N_m(i) \geq u\} \\
&\leq e^{-\lambda u} \left(\mathbf{E}\left\{ e^{\lambda (K_m - i)_+} \right\} \right)^m \\
&\leq e^{-\lambda u} \left(\mathbf{P}\{K_m \leq i\} + \sum_{j=1}^{\infty} e^{\lambda j} \mathbf{P}\left\{K_m = i + j\right\} \right)^m \\
&\leq e^{-\lambda u} \left(1 + \sum_{j=1}^{\infty} e^{\lambda j} \left(1 - \frac{m-1}{n}\right) \left(\frac{m-1}{n}\right)^{i+j-1} \right)^m \\
&\leq e^{-\lambda u} \left(1 + \sum_{j=1}^{\infty} e^{\lambda j} \left(\alpha\right)^{i+j-1} \right)^m \\
&\leq e^{-\lambda u} \left(1 + \alpha^{i-1} \frac{e^{\lambda}\alpha}{1 - e^{\lambda}\alpha} \right)^m \\
&= e^{-\lambda u} \left(1 + \frac{1+\alpha}{1-\alpha} \alpha^{i-1} \right)^m \\
&\qquad \text{(set } e^{\lambda}\alpha = (1+\alpha)/2\text{)} \\
&\leq \exp\left(-u \log((1+\alpha)/2\alpha) + \frac{1+\alpha}{1-\alpha} \alpha^{i-1} m \right) \\
&= \exp\left(-\frac{1+\alpha}{1-\alpha} \alpha^{i-1} m \right) \\
&\qquad \text{(set } u = \tfrac{2(1+\alpha)\alpha^{i-1} m}{(1-\alpha)\log((1+\alpha)/2\alpha)}\text{)}.
\end{aligned}$$

This concludes the proof. □

Note that, for any given R, the cardinality of the R-head in the previous section is not more than $N_m(R)$. Assume that we were to start with an R-head of size $m' \leq \alpha' n$, where R and α' are defined in Lemma 5. Then, by mimicking the argument of the previous section, we have

LEMMA 5. *Define $b = (2\lambda + 2)\alpha'$ and assume that $b < 1$. Define*

$$D = \left\lfloor \log_2 \left(\frac{2}{3\log(1/b)} \right) - 0.1 \right\rfloor ,$$

and

$$R = \lceil \lambda \log\left(\beta(2\lambda + 3)\right) \rceil .$$

Let Z be the number of elements in the $R + r$-head, with $r = \lfloor \log_2 \log n \rfloor + D$. Then

$$\lim_{n\to\infty} \mathbb{P}\left\{ Z \geq \frac{nb^{2^r}}{2\lambda + 2} \right\} = 0 .$$

In particular,

$$\lim_{n\to\infty} \mathbb{P}\left\{ Z \geq n^{1-1/(6\times 2^{0.1})} \right\} = 0 .$$

Note in particular that the only difference between Lemma 2 and Lemma 5 is in the replacement of α in the definition of b by α'. The definition of λ is unaltered. Lemma 3 would then imply that with probability tending to one, $M_n \leq R + r + 2$, with r as in Lemma 5. Since the number of elements in the R-head is random, we use the following argument, based on Lemma 4. Define

$$\beta = \frac{2(1+\alpha)}{(1-\alpha)\log((1+\alpha)/2\alpha)}.$$

Then

$$\begin{aligned}
&\mathbb{P}\{M_n > R + r + 2\} \\
&\quad \leq \mathbb{P}\left\{ N_m(R) \geq \beta m \alpha^{R-1} \right\} + \mathbb{P}\{M_n > R + r + 2 \mid N_m(R) \leq \beta m \alpha^{R-1}\} \\
&\quad \leq \exp\left(-\frac{1+\alpha}{1-\alpha} m \alpha^{R-1} \right) + o(1)
\end{aligned}$$

provided that $\beta m \alpha^{R-1} \leq \alpha' n$ where $\alpha' = 1/(2\lambda + 3)$ (to make b in Lemma 5 less than one). But $\beta m \alpha^{R-1} \leq \beta n \alpha^R$, and thus it suffices to set

$$R = \left\lceil \frac{\log\left(\beta(2\lambda + 3)\right)}{\log(1/\alpha)} \right\rceil = \lceil \lambda \log\left(\beta(2\lambda + 3)\right) \rceil .$$

With this choice of R, and the choice of r given in Lemma 5, we thus conclude that

$$\lim_{n\to\infty} \mathbb{P}\{M_n > R + r + 2\} = 0 .$$

This concludes the proof of Theorem 1.

Proof of theorem 2

We prove the theorem by construction. This is done by identifying a subset of elements that must be of age at least 1, a further subset of age 2, and so forth. We show that this process can be carried out at least k times with high probability until we run out of elements, where k is of the order of $\log\log n$. We consider all values $X_{i,0}$, $1 \le i \le m$ first. Consider that "ball" i is dropped in urn $X_{i,0}$. If an urn receives j elements, then at least $j-1$ of them must move on, and will have an age at least equal to one. Who moves on depends upon the tie-breaking strategy, but in our analysis, it only matters to know how many move on. We introduce A_r, the number of elements that are marked in the r-th step. A_1 is the number of elements of age at least 1 in the process above. We formally set $A_0 = m$. Given A_{r-1}, we take the A_{r-1} elements of age at least $r-1$ (note: these are not the only ones of age at least $r-1$), and look at their $X_{i,r}$ values, with the number of i's clearly being A_{r-1}. We consider the subset that has to move on, so only urns with at least two elements can be of any use. Note that in view of the tie-breaking policy, an earlier element may move on. But in any case, if an urn receives j elements from the A_{r-1}, at least $j-1$ of them must move on and increase their age by one. These $j-1$ elements are collected and form a further subset of size A_r, consisting entirely of elements of age at least r.

We return now to our process A_r. We observe that A_r has the (A_{r-1}, n) urn distribution. The inequalities (1) suggest natural bounds for A_r. We define an integer sequence a_r such that with high probability, $a_r \le A_r$. We have $a_0 = m = \lceil \alpha n \rceil$. Then set

$$a_{r+1} = a_r^2 / 8n \ .$$

Note that

$$a_r = 8n(a_0/8n)^{2^r} \ge 8n(\alpha/8)^{2^r} \ .$$

Define the events

$$E_r = \cap_{j \le r} [a_j \le A_j]$$

and let $(.)^c$ denote the complement of an event. Observe the following:

$$\mathrm{P}\{E_r^c\} \le \mathrm{P}\{E_1^c\} + \sum_{j=2}^{r} \mathrm{P}\{E_j^c \mid E_0, \ldots, E_{j-1}\} = \sum_{j=2}^{r} \mathrm{P}\{E_j^c \mid E_{j-1}\} \ .$$

Also, if r is so small that at all times $a_{r-1} \ge 4$ (a condition that is needed so that we may apply the inequalities derived in the section entitled "balls in urns"), we have

$$\begin{aligned} \mathrm{P}\{E_r^c \mid E_{r-1}\} &\le \mathrm{P}\left\{[A_r < a_r] \mid a_{r-1} \le A_{r-1}\right\} \\ &\le \mathrm{P}\left\{ A_r < \frac{1}{2}\mathrm{E}\{A_r \mid a_{r-1} \le A_{r-1}\} \mid a_{r-1} \le A_{r-1} \right\} \ , \end{aligned}$$

provided that $a_r \le (1/2)\mathrm{E}\{A_r \mid a_{r-1} \le A_{r-1}\}$. But this follows from $a_r = a_{r-1}^2/8n = (1/2)a_{r-1}^2/4n \le \mathrm{E}\{A_r \mid a_{r-1} \le A_{r-1}\}$. We let A have the $(\lceil a_{r-1} \rceil, n)$ urn distribution.

Thus,

$$\begin{aligned}
\mathbf{P}\{E_r^c \mid E_{r-1}\} &\le \mathbf{P}\left\{A < \frac{1}{2}\mathbf{E}\{A\}\right\} \\
&\le \frac{\lceil a_{r-1} \rceil}{2(\mathbf{E}\{A\}/2)^2} \\
&\le \frac{2\lceil a_{r-1} \rceil}{((\lceil a_{r-1} \rceil)^2/4n)^2} \\
&\le \frac{32n^2}{a_{r-1}^3} \\
&\le \frac{(8/\alpha)^{3\times 2^{r-1}}}{16n} .
\end{aligned}$$

Therefore,

$$\mathbf{P}\{E_r^c\} \le \sum_{j=0}^{r-1} \frac{(8/\alpha)^{3\times 2^j}}{16n} = \frac{1}{16n}\sum_{j=0}^{r-1}(8/\alpha)^{3\times 2^j} \le \frac{(8/\alpha)^{3\times 2^{r-1}}}{16n(1-(\alpha/8)^3)} .$$

Set $r = \lceil \log_2(c \log n) \rceil$ for $c > 0$, and note that the upper bound is not more than

$$\frac{n^{3c\log(8/\alpha)-1}}{16(1-\alpha/8)}$$

and this tends to zero if $c < 1/3\log(8/\alpha)$. With that choice of r, we note that

$$a_r \ge \frac{8n}{(8/\alpha)^{3\times 2^r}} \ge \frac{8n}{n^{6c\log(8/\alpha)}} \ge 8$$

provided we take $c = 1/6\log(8/\alpha)$. With such a choice, we then have

$$\mathbf{P}\{A_r = 0\} \le \mathbf{P}\{A_r < a_r\} \le \mathbf{P}\{E_r^c\} \le \frac{1}{16(1-(\alpha/8)^3)\sqrt{n}} . \square$$

References

O. Amble and D. E. Knuth, "Ordered hash tables," *Computer Journal*, vol. 17, pp. 135–142, 1974.

Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, "Balanced allocations," *SIAM Journal on Computing*, vol. 29, pp. 180–200, 1999.

K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Mathematical Journal*, vol. 37, pp. 357–367, 1967.

A.D. Barbour, L. Holst and S. Janson, *Poisson Approximation*, Oxford University Press, 1992.

A. Z. Broder and A. R. Karlin, "Multilevel adaptive hashing," in: *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 43–53, SIAM, Philadelphia, 1990.

A. Z. Broder and M. Mitzenmacher, "Using multiple hash functions to improve IP lookups," in: *INFOCOM 2001*, pp. 0–0, 2001.

A. Brodnik and I. Munro, "Membership in constant time and almost-minimum space," *SIAM Journal on Computing*, vol. 28, pp. 1627–1640, 1999.

P. Celis, P.-Å. Larson, and J. I. Munro, "Robin Hood hashing," in: *26th IEEE Sympusium on the Foundations of Computer Science*, pp. 281–288, 1985.

P. Celis, "Robin Hood hashing," Technical Report CS-86-14, Computer Science Department, University of Waterloo, 1986.

H. Chernoff, "A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations ," *Annals of Mathematical Statistics*, vol. 23, pp. 493–507, 1952.

A. Czumaj and V. Stemann, "Randomized Allocation Processes," in: *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science (FOCS'97), October 19-22, 1997, Miami Beach, FL*, pp. 194–203, 1997.

L. Devroye, "The expected length of the longest probe sequence when the distribution is not uniform," *Journal of Algorithms*, vol. 6, pp. 1–9, 1985.

M. Dietzfelbinger and F. Meyer auf de Heide, "A new universal class of hash functions and dynamic hashing in real time," in: *Proceedings of the 17th International Colloquium on Automata, Languages and Programming (ICALP '90)*, vol. 443, pp. 6–19, Lecture Notes in Computer Science, 1990.

M. Dietzfelbinger, J. Gil, Y. Matias, and N. Pippenger, "Polynomial hash functions are reliable (extended abstract)," in: *Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP '92)*, vol. 623, pp. 235–246, Lecture Notes in Computer Science, 1992.

M. Dietzfelbinger, A. Karlin, K. Mehlhorn, F. Meyer auf de Heide, H. Rohnert, and R. E. Tarjan, "Dynamic perfect hashing: upper and lower bounds," *SIAM Journal on Computing*, vol. 23, pp. 738–761, 1994.

B. Efron and C. Stein, "The jackknife estimate of variance," *Annals of Statistics*, vol. 9, pp. 586–596, 1981.

P. Flajolet, P. V. Poblete, and A. Viola, "On the analysis of linear probing hashing," *Algorithmica*, vol. 22, pp. 490–515, 1998.

M. L. Fredman, J. Komlós, and E. Szemerédi, "Storing a sparse table with O(1) worst case access time," *Journal of the ACM*, vol. 31, pp. 538–544, 1984.

G. H. Gonnet, "Expected length of the longest probe sequence in hash code searching," *Journal of the ACM*, vol. 28, pp. 289–304, 1981.

G. H. Gonnet and R. Baeza-Yates, *Handbook of Algorithms and Data Structures (2nd ed.)*, Addison-Wesley, 1991.

G. R. Grimmett and D. R. Stirzaker, *Probability and Random Processes*, Oxford University Press, 1992.

W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, pp. 13–30, 1963.

D. E. Knuth, *The Art of Computer Programming, Vol. 3 : Sorting and Searching*, 2nd edition, Addison-Wesley, Reading, Mass., 1998.

C. McDiarmid, "On the method of bounded differences," in: *Surveys in Combinatorics*, (edited by J. Siemons), vol. 141, pp. 148–188, London Mathematical Society Lecture Note Series, Cambridge University Press, 1989.

C. McDiarmid, "Concentration," in: *Probabilistic Methods for Algorithmic Discrete Mathematics*, (edited by M. Habib and C. McDiarmid and J. Ramirez-Alfonsin and B. Reed), pp. 195–248, Springer, New York, 1998.

M. Mitzenmacher, "Studying balanced allocations with differential equations," Technical Note 1997024, Digital Equipment Corporation Systems Research Center, Palo Alto, CA, 1997.

M. Mitzenmacher, A. W. Richa, and R. Sitaraman, "The power of two random choices: a survey of techniques and results," Technical Report, 2000.

R. Pagh and F. F. Rodler, "Cuckoo hashing," BRICS Report Series RS-01-32, Department of Computer Science, University of Aarhus, 2001.

P. V. Poblete and J. I. Munro, "Last-Come-First-Served hashing," *Journal of Algorithms*, vol. 10, pp. 228–248, 1989.

J. M. Steele, "An Efron-Stein inequality for nonsymmetric statistics," *Annals of Statistics*, vol. 14, pp. 753–758, 1986.

A. Viola and P. V. Poblete, "Analysis of linear probing hashing with buckets," *Algorithmica*, vol. 21, pp. 37–71, 1998.

J. S. Vitter and P. Flajolet, "Average-case analysis of algorithms and data structures," in: *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, (edited by J. van Leeuwen), pp. 431–524, MIT Press, Amsterdam, 1990.