Distributional Analysis of the Parking Problem and Robin Hood Linear Probing Hashing with Buckets

Alfredo Viola
Instituto de Computación
Universidad
Montevideo, URUGUAY
viola@fing.edu.uy

To Philippe

Motivation

- Simplest collision resolution for open addressing (Peterson '57).
- Works well for tables that are not too full.
- Because of primary clustering, its performance deteriorates when the load factor is high.
- Its analysis leads to nontrivial and interesting mathematical problems. There are connections with tree inversions, tree path lenghts, graph connectivity, area under excursion, etc.
- Equivalent formulation in terms of the parking problem [Knuth].
- First problem that D. Knuth analyzed [Knuth 1963] with bucket size 1, and motivated the collection *"The Art of Computer Programming"*.
- The analysis for general bucket size b presents very interesting challenges. For example, can symbolic methods be used?

[My first analysis of an algorithm, originally done during summer 1962 in Madison.]

8354

NOTES ON "OPEN" ADDRESSING.

D. Knuth 7/22/63

1. Introduction and Definitions. Open addressing is a widely-used technique for keeping "symbol tables." The method was first used in 1954 by Samuel, Amdahl, and Boehme in an assembly program for the IBM 701. An extensive discussion of the method was given by Peterson in 1957 [1], and frequent references have been made to it ever since (e.g. Schay and Spruth [2], Iverson [3]). However, the timing characteristics have apparently never been exactly established, and indeed the author has heard reports of several reputable mathematicians who failed to find the solution after some trial. Therefore it is the purpose of this note to indicate one way by which the solution can be obtained.

We will use the following abstract model to describe the method: N is a positive integer, and we have an array of N variables $x_1, x_2, \ldots, x_N$. At the beginning, $x_i = 0$, for $1 \leq i \leq N$.

To "enter the k-th item in the table," we mean that an integer a_k is calculated, $1 \leq a_k \leq N$, depending only on the item, and the following process is carried out:

1. Set $j = a_k$.
2. "The comparison step." If $x_j = 0$, set $x_j = 1$ and stop; we say "the k-th item has fallen into position x_j."
3. If $j = N$, go to step 5.
4. Increase j by 1 and return to step 2.
5. "The overflow step." If this step is entered twice, the table is full, i.e. $x_i = 1$ for $1 \leq i \leq N$. Otherwise set j to 1 and return to step 2.

Observe the cyclic character of this algorithm.

We are concerned with the statistics of this method, with respect to the number of times the comparison step must be executed. More precisely, we consider all of the N^k possible sequences $a_1, a_2, \ldots a_k$ to be equally probable, and we ask, "What is the probability that the comparison step is used precisely m times when the k-th item is placed?"

2. Non-overflow (self-contained) sequences.

Let $\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right]$ denote the number of sequences $a_1, a_2, \ldots a_k$ $(1 \leq a_i \leq n)$ in which no overflow step occurs during the entire process of placing k items, if the algorithm is used for $N = n$. (By convention, we set

$$\left[\begin{smallmatrix} n \\ 0 \end{smallmatrix}\right] = 1.)$$

Lemma 1: If $0 \leq k \leq n+1$, then $\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right] = (n+1)^k - k(n+1)^{k-1}$.

Proof: This proof is based on the fact that $\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right]$ is precisely the number of sequences $b_1, b_2, \ldots b_k$ $(1 \leq b_i \leq n+1)$ in which, if the algorithm is carried out for $N = n+1$, then $x_{n+1} = 0$ at the end of the operation. This follows because every sequence of the former type is one of the latter, and conversely the condition implies in particular that $1 \leq b_i \leq n$, and that no overflow step occurs.

But sequences of the latter type are easily enumerated, because the algorithm has circular symmetry; of the $(n+1)^k$ possible sequences $b_1, b_2, \ldots b_k$, exactly $k/(n+1)$ of these leave $x_{n+1} \neq 0$. This shows that

$$\left[\begin{smallmatrix} n \\ k \end{smallmatrix}\right] = (n+1)^k \left(1 - \frac{k}{n+1}\right).$$

Collision Resolution Strategies

- In open addressing, when two keys collide, either one of them may stay in that location, while the other one keeps probing.

Some heuristics:

- *"First-Come-First-Served" (standard)*.
  - Each collision is resolved in favor of the first record that probed the location.
- *Last-Come-First-Served* [Poblete and Munro '89].
  - Each collision is resolved in favor of the incoming record.
- *Robin Hood* [Celis, Larson and Munro '85].
  - Each collision is resolved in favor of the record that is further away from its home location.

An example of Robin Hood Linear Probing Hashing (b = 2)

Keys inserted:

- 36, 77, 24, 79, 56, 69, 49, 18, 38, 97, 78, 10, 58.

Hash function

- $h(x) = x$ mod 10.

a

49	79	40		24		36	77	18	58
69	10	70				56	97	38	78
0	1	2	3	4	5	6	7	8	9

What happens when 29 is inserted?

a

29	69	10	70	24		36	77	18	58
49	79	40				56	97	38	78
0	1	2	3	4	5	6	7	8	9

Properties of Robin Hood Linear Probing Hashing

a

29	69	10	70	24		36	77	18	58
49	79	40				56	97	38	78
0	1	2	3	4	5	6	7	8	9

- At least one record is in its home bucket.
- The keys are stored in nondecreasing order by hash value, starting at some location k and wrapping around. In our example, $k = 5$ (the first slot of the third bucket).
- If a fixed rule is used to break ties among the candidates to probe their next probe bucket (eg: by sorting these keys in increasing order), then the resulting table is independent of the order in which the records were inserted. Then, *we may insert the elements in any order, and study the behavior of the last one inserted*.

The exact filling model and the Poisson model

- *Exact filling model.*
  - A fixed number of keys, *n*, are distributed among *m* locations, and all m^n possible arrangements are equally likely to occur.

- *Poisson model.*
  - Each location receives a number of keys that is Poisson distributed with parameter $b\alpha$, and is **independent** of the number of keys going elsewhere. This implies that the total number of keys, *N*, is itself a Poisson distributed random variable with parameter $b\alpha m$:

$$Pr[N = n] = \frac{e^{-b\alpha m}(b\alpha m)^n}{n!}.$$

The Poisson Transform

- Results in one model can be transfered into the other model by the *Poisson Transform*:

$$P_m[f_{m,n}; b\alpha] = \sum_{n \geq 0} Pr[N = n] f_{m,n} = e^{-b\alpha m} \sum_{n \geq 0} \frac{(b\alpha m)^n}{n!} f_{m,n}.$$

- *Inversion Theorem*

$$\text{If } \mathbf{P}_m[f_{m,n}; b\alpha] = \sum_{k \geq 0} a_{m,k} (bm\alpha)^k \text{ then } f_{m,n} = \sum_{k \geq 0} a_{m,k} \frac{n^{\underline{k}}}{(bm)^k}.$$

where $n^{\underline{k}} = n(n-1)\ldots(n-k+1)$.

- The results obtained under the Poisson filling model can also be interpreted as an approximation of those one would obtain under the exact filling model when $n, m \to \infty$ with $n/m = b\alpha$ with $0 \leq \alpha < 1$.

Combinatorial interpretation of Linear Probing

a

29	69	10		24		36	77	18	58
49	79					56	97	38	78
0	1	2	3	4	5	6	7	8	9

- Any Linear Probing Hash table can be seen as a sequence of *almost full* tables, where an "almost full" table is a subtable with all but the last bucket full [Knuth 1997; Flajolet, Poblete & V. 1997].

- **Example:** [3-3],[4-4],[5-5],[6-2].

- This interpretation can be nicely handled by Analytic Combinatorics, since for example, it implies that it is enough to study almost full tables, and then use the "sequence" construction.

Some Previous Work in Linear Probing (b=1)
Distributional Analysis

Individual Displacements

- [Knuth 1963]. First nontrivial algorithm he analyzed.
- [Konhein & Weiss 1966] First published analysis.
- [Janson 2006; V. 2006]. Distributional analysis for the LCFS, FCFS and RH heuristics.

Construction cost

- [Knuth 1997; Flajolet, Poblete & V. 1997]. Distributional analysis and relations with graph connectivity, tree inversions, tree path lenghts, area under excursions, etc. *(Airy Phenomena)*.
- [Janson 2001]. More limit distributions.
- [Chassaing - Lourchard 2002]. Phase Transitions.

Previous Work in Linear Probing (b ≥ 1)

- [Blake & Konheim 1977]. Exact PGF for the number of ways of constructing a hash table in the Exact Filling model. Asymptotic results of the expected cost of the displacement of a random element for an α-full table ($\alpha < 1$) using the FCFS heuristic.

- [Poblete & V. 1998]. Exact expected value for the search cost of a random element and asymptotic results when the table is full ($\alpha = 1$).

- [Knuth - volume 3]. Expected value for the search cost of a random element for α-full tables ($\alpha < 1$).

- However there has not been any *distributiononal analysis* ...

Some Previous Work in Parking Problem (b = 1)

- The problem has been extensively studied.
- The problem was first proposed in [Konheim & Weiss, 1966].
- Latest results related with phase transitions and limit distributions are [Chassaing & Marckert, 2001], [Chassaing & Louchard, 2002], [Panholzer, 2008].
- In [Prellberg, Cameron, Johannsen & Schweitzer, 2008] a distributional analysis in the exact filling model is presented. Some of these results can be rederived in an independent way by depoissonization of the results in the Poisson Model presented in [Gonnet & Munro 1984, V. 2006].
- Nevertheless there has been no published results for buckets with $b > 1$.
- In his diploma thesis, [Seitz 2009], has independently presented some of the results presented here.

Our contributions

- Distributional analysis for the *search cost of a random element using the RH Heuristic* under a Poisson model ($\alpha < 1$).
- Exact results follow by *depoissonization*.
- Distributional analysis for the *bucket occupancy* in the exact and asymptotic models, based on a new sequence of numbers.
- Distributional analysis for the *parking problem follows* from the analysis for the RH heuristic.

Bucket Occupancy

- Let $Q^b_{m,n,d}$ count the number of ways to insert n records in a table with m buckets of size b so that a given bucket (say the last one) has *more* than d empty slots.

- Then $\frac{Q^b_{m,n,b-d+1} - Q^b_{m,n,b-d}}{m^n}$ is the *probability* that a given bucket has *exactly* d elements.

- This sequence satisfies the following recursive relation:

$$Q^b_{m,n,d} = \begin{cases} \sum_{j=0}^{n} \binom{n}{j} Q^b_{m-1,j,d} & 0 \leq n < mb - d \\ 0 & n \geq mb - d \end{cases},$$

with the border condition,

$$Q^b_{0,n,d} = [n = 0].$$

Bucket Occupancy (cont.)

- There is a direct translation into exponential generating functions:

$$\begin{aligned} Q^b_{0,d}(b\alpha) &= 1 \\ Q^b_{m,d}(b\alpha) &= \left[e^{b\alpha} Q^b_{m-1,d}(b\alpha)\right]_{bm-d-1} \qquad m \geq 1, \end{aligned}$$

  where

$$Q^b_{m,d}(b\alpha) = \sum_{n \geq 0} Q^b_{m,n,d} \ \frac{(b\alpha)^n}{n!}.$$

- We have truncated generating functions!

New sequence of numbers

- Find an exponential generating function $T_d^b(b\alpha)$ such that

$$Q_{m,d}^b(b\alpha) \quad = \quad \left[T_d^b(b\alpha)e^{bm\alpha}\right]_{bm-d-1}$$

with

$$T_d^b(b\alpha) = \sum_{k\geq 0} T_{k,d}^b \quad \frac{(b\alpha)^k}{k!},$$

for some coefficients $T_{k,d}^b$ to be determined, and independent of m.

- The coefficients of these generating functions satisfy

$$Q_{m,n,d}^b = \sum_{k\geq 0} \binom{n}{k} T_{k,d}^b \, m^{n-k} \qquad 0 \leq n < mb - d.$$

New sequence of numbers (cont.)

- The following theorem can be used as a definition of these numbers.

  Theorem (Poblete & V. 1998)

$$\sum_j \binom{k}{j} \left(\left\lfloor \frac{k+d}{b} \right\rfloor \right)^{k-j} T^b_{j,d} = [k=0].$$

- An important property is

$$\sum_{d=0}^{b-1} T^b_{k,d} = \begin{cases} b & \text{k} = 0 \\ -1 & \text{k} = 1 \\ 0 & \text{k} > 1. \end{cases}$$

- **Theorem (Bucket Occupancy in the Poisson Model)**

$$\lim_{m\to\infty} \mathcal{P}_m \left[\frac{Q^b_{m,n,d}}{m^n}; b\alpha \right] = T^b_d(b\alpha).$$

New sequence of numbers (cont.)

- **Lemma**

$$\mathbf{P}_m[Q_{m,n,d}/m^n; b\alpha] = T_d(b\alpha) + O\left((b\alpha)^{bm-d}\right).$$

- **Theorem (Bucket Occupancy)**

  Let $\Upsilon_{m,d}(b\alpha) = \mathbf{P}_m[Q_{m,n,d}/m^n; b\alpha]$. That is, $\Upsilon_{m,d}(b\alpha)$ is the **probability, in the Poisson Model**, that a given bucket contains more than d empty slots when $bm\alpha$ elements are inserted in a hash table with m buckets of size b, using linear probing as collision resolution scheme. Then, when $\alpha < 1$,

$$\Upsilon_{m,d}(b\alpha) = T_d(b\alpha) + O\left(\alpha^{-d} e^{bm(1-\alpha+\log\alpha)} m^{-3/2}\right).$$

- The rate of convergence to this limit value is exponentially small.

The family $T^b_{k,d}$

- $T^1_{k,0} \rightarrow$ 1, -1, 0, 0, 0, ...

- $T^2_{k,0} \rightarrow$ 1, 0, -1, 2, -8, 48, -378, 3672, -42368, ...
- $T^2_{k,1} \rightarrow$ 1, -1, 1, -2, 8, -48, 378, -3672, 42368, ...

- $T^3_{k,0} \rightarrow$ 1, 0, 0, -1, 3, -6, -12, 264, -2016, ...
- $T^3_{k,1} \rightarrow$ 1, 0, -1, 2, -3, -2, 60, -408, 1341, ...
- $T^3_{k,2} \rightarrow$ 1, -1, 1, -1, 0, 8, -48, 144, 675, ...

- The sequence $T^2_{k,0} \rightarrow$ is sequence *A124453* in Sloane's Encyclopedia.
- $T^b_0(b\alpha)$ is studied in [Blake & Konheim 1977].

Generating Functions

- If we generalize the derivations presented in [Blake & Konheim 1977], based on a *combinatorial interpretation* of the problem we have

  Theorem (V. 2010)

$$T_d^b(b\alpha) = \left[z^d\right] b(1-\alpha) \frac{\prod_{j=1}^{b-1}\left(1 - z\,\frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)},$$

  where $r = e^{\frac{2\pi \mathbf{i}}{b}}$ is a b-th root of unity, and $T(u) = u\exp(T(u))$ is the Tree function.

- **Corollary**

$$\sum_{d=0}^{b-1} T_{b-1-d}^b(b\alpha) z^d = b(1-\alpha) \frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}.$$

Generating Functions (cont.)

- Similar generating functions can be found for $Q^b_{m,d}(z)$, based on symbolic methods!

- **Corollary**

$$Q^b_{m,d}(b\alpha) = \left[\left[z^d\right] b(1-\alpha) \frac{\prod_{j=1}^{b-1} \left(1 - z \, \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1} \left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)} e^{bm\alpha} \right]_{bm-d-1} .$$

Combinatorial Interpretation of results in Blake & Konheim 1977

a

29	69	10	70	24		36	77	18	58
49	79	40				56	97	38	78
0	1	2	3	4	5	6	7	8	9

- A Linear Probing Hash table is a sequence of *almost full* tables.
- **Example:** [3-3],[4-4],[5-5],[6-2].
- Let F_{bn+d} be the number of ways to construct an almost full table of length $n+1$ and size $bn+d$ (that is, there are $b-d$ empty slots in the last bucket).

$$F_d(u) = \sum_{n\geq 0} F_{bn+d} \frac{u^{bn+d}}{(bn+d)!} \qquad N_d(z,w) = \sum_{s=0}^{b-1-d} w^{b-s} F_s(zw), \quad 0 \leq d \leq b-1.$$

- $N_d(z,w)$ is the generating function for the number of almost full tables with more than d empty locations in the last bucket.

Combinatorial Interpretation of results in Blake & Konheim 1977 (cont.)

- The *elementary symmetric functions* of variables $\gamma_j(z)$ are the coefficients $\{\sigma_k(z)\}$ given by $\sum_{k=0}^{b}\sigma_k(z)x^{n-k}=\prod_{j=0}^{b-1}(x+\gamma_j(z))$.

- Let r a primitive b-th root of unity and $\sigma_k(z)$ be the k-th elementary symmetric function of the variables $\{T(r^jz), 0\le j<b\}$, where T is the Tree function. Lemma 2.3 (pag. 594) states

$$(bz)^{b-d}F_d(bz)=(-1)^{b-d-1}b^{b-d}\sigma_{b-d}(z),$$

  and formula 3.8 (pag. 597) states

$$N_0(z,w)=1-\prod_{i=0}^{b-1}\left(1-\frac{b}{z}T\left(r^i\frac{zw}{b}\right)\right).$$

- **Theorem (V. 2010)**

$$N_d(b\alpha,e^{-\alpha})=[u^d]\prod_{i=1}^{b-1}\left(1-u\frac{T(r^i\alpha e^{-\alpha})}{\alpha}\right),\quad 0\le d\le b-1.$$

Combinatorial Interpretation of results in Blake & Konheim 1977 (cont.)

- Let $$\Lambda_d(z,w) = \sum_{m\geq 0}\sum_{n\geq 0} Q_{m,n,d}\frac{z^n}{n!}w^{bm}.$$

- $\Lambda_0(z,w)$ is the generating function for the number of ways to construct hash tables such that their last bucket is not full. After a somehow tedious calculation, Lemma 3.2 (page 597) states

$$\Lambda_0(z,w) = \frac{N_0(z,w)}{1-N_0(z,w)} = \frac{1}{\prod_{i=0}^{b-1}\left(1-\frac{b}{z}T\left(r^i\frac{zw}{b}\right)\right)} - 1.$$

- **Combinatorial interpretation!:** Squence of almost full tables!

- **Generalization (V. 2010):**

$$\Lambda_d(z,w) = \frac{N_d(z,w)}{1-N_0(z,w)}.$$

Combinatorial Interpretation of results in Blake & Konheim 1977 (cont.)

- After a far from trivial analysis, Theorem 4.1 states

$$T_0(b\alpha) = \frac{b(1-\alpha)}{\prod_{j=1}^{b-1}\left(1-\frac{T\left(r^j\alpha e^{-\alpha}\right)}{\alpha}\right)},$$

where T is the Tree function and r is a b-th root of unity.

- **Generalization** based on combinatorial interpretation!

$$T_d(b\alpha) = N_d(b\alpha, e^{-\alpha})T_0(b\alpha), \quad 0 \le d \le b-1.$$

- **Theorem (V. 2010)**

$$T_d^b(b\alpha) = \left[z^d\right] b(1-\alpha)\,\frac{\prod_{j=1}^{b-1}\left(1-z\,\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}, \quad 0 \le d \le b-1.$$

Parking Problem

a

29	69	10	70	24		36	77	18	58
49	79	40				56	97	38	78
0	1	2	3	4	5	6	7	8	9

- For $b = 1$, when the hash function is order preserving, Robin Hood linear probing can be used to sort (Gonnet and Munro '84).
- Instead of letting the excess records from the rightmost bucket of the table wrap around to bucket zero, we can use an overflow area consisting of buckets m, $m + 1$, etc.
- Without loss of generality we search for a record that hashes to bucket 0.
- To search for a random element that hashes to 0, we have to probe locations occupied by the elements that would have gone to the overflow area. This is the *parking problem*.
- We then have to consider collisions with all the other elements that hash to 0.

Analysis of the overflow area (Parking Problem)

- Let $w_{m,b\alpha,k}$ be the probability of having k cars going to overflow from bucket m in a $b\alpha$-full table of size m.

- Let $\Omega(m, b\alpha, z) = \sum_{k\geq 0} w_{m,b\alpha,k} z^k$ be the probability generating function for the number of elements that overflow from bucket m.

- With probability $e^{-b\alpha}\frac{(b\alpha)^k}{k!}$ bucket m receives, in addition to the elements that overflow from the previous bucket, k elements that hash to it. From these elements, all but b of them go to overflow, and their contribution to the recurrence is

$$\Omega_m(b\alpha, z) \approx e^{b\alpha(z-1)}\frac{\Omega_{m-1}(b\alpha, z)}{z^b}.$$

- However, when this bucket receives less than b elements, there is no overflow, and so we need a correction term $\sum_{s=1}^{b}(1 - z^{-s})\, P_{m,s}(b\alpha)$, where $P_{m,s}(b\alpha)$ is the probability of having $b - s$ elements in a given bucket.

Parking Problem (cont.)

- The contribution of this correction term is

$$\sum_{s=1}^{b}\left(1-z^{-s}\right)\left(\Upsilon_{m,s-1}(b\alpha)-\Upsilon_{m,s}(b\alpha)\right)=\left(1-z^{-1}\right)\sum_{s=0}^{b-1}\Upsilon_{m,s}(b\alpha)z^{-s}.$$

- As a consequence we obtain the following recurrence:

$$\Omega_m(b\alpha,z)=e^{b\alpha(z-1)}\frac{\Omega_{m-1}(b\alpha,z)}{z^b}+\left(1-z^{-1}\right)\sum_{s=0}^{b-1}\Upsilon_{m,s}(b\alpha)z^{-s}.$$

that leads to

$$\Omega_m(b\alpha,z)=\left(\frac{b(1-\alpha)(z-1)}{z^b-e^{b\alpha(z-1)}}\right)\frac{\prod_{j=1}^{b-1}\left(z-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1-\frac{T(r^j\alpha e^{-\alpha})}{\alpha}\right)}+O\left(\alpha^{bm}\right).$$

Parking Problem (cont.)

- **Theorem**

  For all $k \geq 0$ we have

$$w_{m,b\alpha,k} = \sum_{j=0}^{\lfloor \frac{k}{b} \rfloor} e^{b\alpha(j+1)} \sum_{i=0}^{\min(b-1,k)} \frac{(-1)^{k-i-bj}}{(k-i-bj)!} (b\alpha(j+1))^{k-i-1-bj}(k-i+b\alpha(j+1)-bj)T_{b-1-i}(b\alpha) + O\left(\alpha^{bm}\right).$$

- **Theorem**

  Let $\Omega_{m,b\alpha}$ be the r.v for the number of cars that overflow from a $b\alpha$-full table with m buckets of size b and $\alpha < 1$. Then

$$\mathbf{E}[\Omega_{m,b\alpha}] = \frac{1}{2}\left(\frac{1}{1-\alpha} - b(1+\alpha)\right) + \sum_{d=1}^{b-1} \frac{1}{\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)} + O\left(\alpha^{bm}\right).$$

Parking Problem (cont.)

- After depoissonization we may prove the following

Corollary

Let $\Omega_{b,m,n}$ be the R.V for the number of cars that overflow from a hash table of length m and size n with buckets of size b. Then

$$\mathbf{E}[\Omega_{b,m,n}] = \sum_{i=2}^{\lfloor n/b \rfloor} \binom{n}{i} \frac{(-1)^i}{m^i} \sum_{k=1}^{m} k^{i-1} \binom{bk-i}{bk-1},$$

$$\mathbf{E}[\Omega_{b,m,bm-1}] = \frac{\sqrt{2\pi bm}}{4} - \frac{7}{6} + \sum_{d=1}^{b-1} \frac{1}{1 - T\left(e^{\frac{2\pi \mathbf{i} d}{b}} - 1\right)} + \frac{1}{48}\sqrt{\frac{2\pi}{bm}} + O\left(\frac{1}{bm}\right).$$

Analysis of Robin Hood Linear Probing Hashing

- Let $\Psi_m(b\alpha, z)$ be the probability generating function for the displacement (number of buckets) of a random record in a $b\alpha$-full table of size m with $0 \leq \alpha < 1$.

- We first derive the generating function $C_m(b\alpha, z)$ for the total displacement, without considering the fact we have to count only number of buckets probed.

- Then, if $C_m(b\alpha, z) = \sum_{n\geq 0} c_{m,n}(b\alpha) z^n$, we have

$$\begin{aligned}
\Psi_m(b\alpha, z) &= \sum_{n\geq 0} c_{m,n}(b\alpha) z^{\lfloor \frac{n}{b} \rfloor} = z \sum_{k\geq 0} \left(\sum_{d=0}^{b-1} c_{m,bk+d}(b\alpha) \right) z^k \\
&= \frac{1}{b} \sum_{d=0}^{b-1} C_m\left(b\alpha, r^d z^{1/b}\right) \sum_{p=0}^{b-1} \left(r^d z^{1/b}\right)^{-p},
\end{aligned}$$

where $r = e^{\frac{2\pi \mathrm{i}}{b}}$ is a b-th root of unity.

Analysis of Robin Hood Linear Probing Hashing (cont.)

- If k elements collide with the searched one, the expected total displacement originated by these collisions for (separately) retrieving all these records is

$$\frac{1}{k+1}\sum_{r=0}^{k} z^r = \frac{1}{k+1}\left(\frac{1-z^{k+1}}{1-z}\right).$$

- Since the probability of having k records colliding with the searched one is $e^{-b\alpha}\frac{(b\alpha)^k}{k!}$, the probability generating function of the displacement originated by these collisions is

$$\frac{e^{-b\alpha}}{1-z}\sum_{k\geq 0}\frac{(b\alpha)^k}{(k+1)!}\left(1-z^{k+1}\right) = \frac{1-e^{b\alpha(z-1)}}{b\alpha(1-z)}$$

Analysis of Robin Hood Linear Probing Hashing (cont.)

- To conclude the derivation we have to consider the cost originated by the elements that overflow:

$$C_m(b\alpha, z) = \frac{b(1-\alpha)(1-e^{b\alpha(z-1)})}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)} \frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)} + O\left(\alpha^{bm}\right).$$

- When $b = 1$, $T_0(\alpha) = (1-\alpha)$, and so we obtain

$$C_m(\alpha, z) = \frac{1-\alpha}{\alpha} \frac{1-e^{\alpha(z-1)}}{z - e^{\alpha(z-1)}} + O\left(\alpha^m\right),$$

as derived in [Janson 2006; V. 2006].

Main Theorem

Theorem

Let $\Psi_{m,b\alpha}$ be the random variable for the cost of searching a random element in a $b\alpha$-full table with m buckets of size b and $\alpha < 1$, using the Robin Hood linear probing hashing algorithm, and let $\Psi_m(b\alpha, z)$ be its probability generating function. Then

$$\begin{aligned}
\Psi_m(b\alpha, z) &= \frac{z}{b}\sum_{d=0}^{b-1} C_m\left(b\alpha, e^{\frac{2\pi \mathrm{i} d}{b}} z^{1/b}\right) \sum_{p=0}^{b-1}\left(e^{\frac{2\pi \mathrm{i} d}{b}} z^{1/b}\right)^{-p}, \quad \text{with} \\
C_m(b\alpha, z) &= \frac{1-e^{b\alpha(z-1)}}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)} \sum_{s=0}^{b-1} T_{b-1-s}(b\alpha) z^s + O\left(\alpha^{bm}\right) \\
&= \frac{b(1-\alpha)(1-e^{b\alpha(z-1)})}{b\alpha\left(z^b - e^{b\alpha(z-1)}\right)} \frac{\prod_{j=1}^{b-1}\left(z - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)} + O\left(\alpha^{bm}\right).
\end{aligned}$$

The distribution

Theorem

The probability $\psi_i(b\alpha)$ that $i+1$ buckets have to be probed to retrieve a random element is

$$\psi_i(b\alpha) = Pr\{\Psi_{b\alpha} = i+1\} =$$

$$\sum_{s=0}^{b-1} \frac{T_{b-1-s}(b\alpha)}{b\alpha} \sum_{k\geq 1} e^{-kb\alpha} \sum_{d=0}^{b-1} \left(\frac{(kb\alpha)^{b(i+k)+d-s}}{(b(i+k)+d-s)!} - \frac{(kb\alpha)^{b(i+k+1)+d-s}}{(b(i+k+1)+d-s)!} \right)$$

$$+O\left(\alpha^{bm}\right).$$

Expected Displacement

Theorem

Let $\Psi_{b,m,n}$ be the random variable for the cost of searching a random element when we insert $n+1$ elements in a hash table of m buckets size b using the Robin Hood linear probing hashing algorithm. Then for $0 \leq \alpha < 1$,

$$\mathbf{E}[\Psi_{b,m,n+1}] = \sum_{k=1}^{\lfloor n/b \rfloor} \sum_{i=kb}^{n} (-1)^{i-kb} \binom{i-1}{kb-1} \frac{(kb)^i}{(i+1)!} \frac{n^{\underline{i}}}{(bm)^i},$$

$$\mathbf{E}[\Psi_{b\alpha}] = \frac{1}{2}\left(\frac{1}{b(1-\alpha)} - 1\right) + \frac{1}{2b\alpha} \sum_{d=1}^{b-1} \frac{\frac{T(r^d \alpha e^{-\alpha})}{\alpha}}{\left(1 - \frac{T(r^d \alpha e^{-\alpha})}{\alpha}\right)},$$

$$\mathbf{E}[\Psi_{b\alpha}] = 1 + \frac{1}{b\alpha}\left(\frac{1}{2}\left(\frac{1}{1-\alpha} - b(1-\alpha)\right) + \sum_{d=1}^{b-1} \frac{1}{\left(1 - \frac{T(r^j \alpha e^{-\alpha})}{\alpha}\right)}\right),$$

$$b\mathbf{E}[\Psi_{b,m,bm-1}] = \frac{\sqrt{2\pi bm}}{4} - \frac{2}{3} + \sum_{d=1}^{b-1} \frac{1}{1 - T\left(e^{\frac{2\pi \mathbf{i} d}{b} - 1}\right)} + \frac{1}{48}\sqrt{\frac{2\pi}{bm}} + O\left(\frac{1}{bm}\right),$$

where $T(u)$ is the Tree Function.

Higher moments

Theorem

$$
\begin{aligned}
E[\Psi^r_{m,b\alpha}] &= r\frac{1-\alpha}{\alpha}\sum_{n\geq 1}(n+1)^{\underline{r-1}}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{b(k+n)+d}}{(b(k+n)+d)!} \\
&- r(r-1)\sum_{n\geq 1}n^{\underline{r-2}}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{b(k+n)+d}}{(b(k+n)+d)!} \\
&\qquad \sum_{s=0}^{b-1-d}\frac{T_{b-1-s}(b\alpha)}{b\alpha} \\
&+ 1^{\underline{r-1}}r\sum_{d=0}^{b-1}\sum_{k\geq 1}e^{-kb\alpha}\frac{(kb\alpha)^{bk+d}}{(bk+d)!}\sum_{s=b-d}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha} \\
&+ 1^{\underline{r}}\sum_{s=0}^{b-1}\frac{T_{b-1-s}(b\alpha)}{b\alpha}\sum_{k\geq 1}e^{-kb\alpha}\sum_{d=0}^{b-1}\frac{(kb\alpha)^{bk+d-s}}{(bk+d-s)!}+O\left(\alpha^{bm}\right).
\end{aligned}
$$

Thank you very much Philippe!