arXiv:2603.10930v1 [cs.DS] 11 Mar 2026

哈夫曼桶草图—一种用于基数估计的简单 $O(m)$算法

Matti Karppa哥德堡大学，瑞典 查尔姆斯理工大学，瑞典

摘要

我们提出哈夫曼桶草图（*HBS*），一种简单且可合并的数据结构，它将具有 m 个寄存器的 HyperLogLog（HLL）草图无损压缩至最优空间 $O(m+\log n)$位，并支持摊销常数时间更新；它可作为HLL的直接替代，在保留可合并性的同时大幅降低内存需求。我们将寄存器划分为小桶，并使用由高度集中的HLL秩分布导出的全局霍夫曼码本对其取值进行编码，同时利用当前基数估计来确定该分布的众数。我们证明，霍夫曼树在整个数据流上仅需重建 $O(\log n)$次，大致在基数翻倍时进行。该框架可扩展到其他具有类似高度集中分布的草图。我们提供了初步数值证据，表明HBS具有实用性，并有望在实践中与当前最先进方法相竞争。

1 引言

在海量数据流中估计不同元素的数量，即基数，是数据库 [Tru19, CDP+19, FN19, WCLL21, PCT23], 网络 [CCD14, BEF+18, DJJ+22, CSGH23], 以及宏基因组学 [BL19, MSPK19], 等领域的一项标准基础操作。由于元素的全集可能实际上是无限的，例如所有变长字符串的集合，假设数据流的真实基数为 n，根据鸽巢原理，为每个元素存储一个代表需要 $\Omega(n \log n)$ 位。即使我们能为全集中的所有元素初始化二进制计数器，要进行精确计数也需要 $\Theta(n)$ 位 [AMS99], ，这可能大到难以实现。不过，使用更少的位也可以估计基数，而且已知 $\Theta(m)$ 位是实现 $\sqrt{\mathrm{Var}\left[\frac{n}{\hat{n}}\right]} = O(1/\sqrt{m})$ 的相对标准误差所必需且充分的 [KNW10]。

不同元素问题的一种实用标准解法是 HyperLogLog（HLL）草图[FFGM07], ，对于相对误差 $O(1/\sqrt{m})$，其大小为 $O(m \log\log n)$ 位。它很有吸引力，因为实现几乎不费吹灰之力，能够以极高的效率实现，支持常数时间更新，而且可合并；这意味着，只需为各个子流维护草图，之后再将它们合并成一个草图，就能轻松以分布式方式处理大型数据流，就像整个数据流由单个草图处理一样。

许多旨在降低 HLL 空间占用的改进工作，不得不牺牲其中一些特性，例如可合并性 [PWY21]或真正的常数时间更新 [KP22]。本文提出了一种简单的算法及配套的数据结构哈夫曼桶草图（*HBS*），利用霍夫曼编码，能够将无损压缩HLL 草图至 $O(m)$ 位，同时保持可合并性并实现高效更新。虽然更新并非真正的常数时间，但其摊销时间为 $O(1)$；在关于 n 与 m之间关系的温和假设下，大多数更新都能在常数时间内完成，只有一小部分更新需要花费更多时间来更新码字。

该算法的核心思想是利用 HLL 草图中寄存器值（或秩）的分布高度集中在 $\lceil \log_2(n/m) \rceil$附近，且尾部衰减极快这一事实。事实上，众所周知，寄存器值的熵渐近地保持为每个寄存器常数 [Dur04], ；这意味着，如果我们将寄存器划分为大小为 $O(\log n)$ 的桶，那么每个桶中的霍夫曼码字长度分布也会高度集中，因此以高概率而言，每个桶只需要 $O(\log n)$ 位。

我们证明，在处理由 n 个不同元素组成的数据流时，只需重建霍夫曼树 $O(\log n)$ 次；本质上，只有当基数翻倍时才需要重建，而我们可以通过维护一个局部基数估计来检测这种情况。也就是说，尽管真实基数 n 未知，我们仍能像 Baron von Münchhausen 一样，揪着自己的头发把自己从沼泽中拉出来，方法是使用

当前的全局估计值来近似寄存器值分布。对于固定数量的寄存器，该分布由基数唯一确定，继而也唯一确定霍夫曼树。

所得数据结构支持基数估计、向草图中插入新元素、查询寄存器值以及合并两个草图等常规操作。草图的渐近大小为 $O(m + \log n)$ 位，这在 [KNW10], 意义下是最优的；插入操作的摊销时间为 $O(1)$，而合并可在 $\tilde{O}(m)$ 时间内完成。[1]

分桶的构想源于实际考量：如果桶足够小，例如只有一个机器字或一个缓存行，那么对其执行的任何操作基本上都是常数时间。我们在理论渐近分析的基础上，进一步探讨该算法在实际应用中的影响及实现方面的考量，并通过数值证据表明，该算法具有与当前最先进方法相竞争的性能。尤其值得一提的是，它实际上可以直接替代 HLL，并且随时可以解压缩回 HLL 草图，因此能够保持相同的可合并性。本文不分析估计过程，因为可以使用任何兼容 HLL 的估计量，例如原始 HLL 估计量、Ertl 的 MLE 估计量 [Ert17],，或 Pettie 和 Wang 提出的当前最先进估计量 [PW21]，其相对方差为 $1.075/m$。

本文其余部分安排如下。第2节回顾符号、数学预备知识、泊松模型，并简述 HLL。第3节是主要结果部分，介绍 HBS 算法和数据结构。第4节在泊松化球与箱模型中对该算法进行渐近分析。第5节简要概述该框架的扩展。第6节讨论实现方面的考量和经验启示，包括草图的内存-方差乘积（MVP）。

1.1 相关工作

Flajolet 和 Martin 开创了通向 HLL 及其变体的一系列研究，他们提出了概率计数算法 [FM85], ，其基础是一个稀疏指示符矩阵，即*FM85* 矩阵。该矩阵的列对应哈希值的几何分布秩，行则对应均匀分布的子流。随后，Durand 和 Flajolet 对此进行了改进，提出 LogLog 算法 [DF03],，其关键创新是只保留每行的最大秩，从而压缩 FM85 矩阵。HLL [FFGM07] 在此基础上以调和平均数取代估计量中的算术平均数，提高了估计量的准确性和稳健性，此后逐渐成为实践中基数估计的标准方案。

HLL 经历了大量实用和理论上的改进与变体，包括 HyperLogLog++[HNH13], 、HLL-Tailcut [XCZL20], 和 HyperLogLogLog [KP22]。另一种相关且简单的基数估计方法是线性计数 [WZT90], ，它可以看作使用一个哈希函数的布隆过滤器；在假设 $m = \Theta(n)$ 的情况下，它对小基数是最优的，并被用作 HLL 中的小基数校正。

Kane、Nelson 和 Woodruff [KNW10], 提出存在一种最优的 $O(m + \log n)$ 位草图，具有常数时间更新；Błasiok [Bla20], 随后对其进行了改进，但这种草图被广泛认为不实用。Pettie 和 Wang 为基数估计确立了精确的信息论极限 [PW21] ，并指出 HLL 草图丢失了 FM85 矩阵中的部分信息，而这些信息有助于进一步压缩，从而证实了 Lang [Lan17] 的观察：FM85 矩阵可以压缩到比 HLL 草图的熵更低。这促成了其他多项研究，它们利用 FM85 矩阵中的额外信息进一步扩展压缩，包括 UltraLogLog [Ert24],、ExaLogLog [Ert25], 和 SpikeSketch [DHS+23]。另一条研究路线则通过使用鞅变换 [PWY21], 放弃了合并能力。最后，虽然我们只讨论草图的压缩，但许多其他研究也着重改进估计器 [Ert17, Ert24, WP23]。

2 预备知识

2.1 数学符号与预备知识

我们用 $[n] = \{1, 2, \ldots, n\}$ 表示前 n 个正整数构成的集合，并用艾弗森括号记法$[\![P]\!]$ 表示谓词 P 的指示符：若 P 为真，则取值为 1；否则取值为 0。

[1] $\tilde{O}(\cdot)$ 隐去多对数因子。

2.2 泊松化球与箱模型

算法的分析是在泊松化的球与箱模型中进行的；这是分析此类算法的标准模型 [Ert17, Ert24, Ert25, PW21, PWY21, WP23]，，且已知它与精确模型几乎等价。

在球与箱模型中，我们将 n 个球独立且均匀地随机投入 m 个箱子，并分析其分布。这种模型的缺点是箱子之间存在依赖关系。为解决这个问题，我们将模型泊松化：让球以 $\lambda = \frac{n}{m}$ 的速率独立、随机地到达各个箱子，这意味着每个箱子中的球数都独立同分布于 Pois(λ)。这使得 n 成为一个随机变量，但假设箱子数量足够大，其分布已知几乎相等。[2] 在实际分析中，我们有两个结合在一起的过程：FM85 矩阵的行是均匀随机选取的，而列则按照几何分布选取。尽管如此，同样适用泊松化技巧。

2.3 HyperLogLog（HLL）

令 w 为字长。我们假设所有哈希函数 $h : \mathcal{U} \to [2^w]$ 都是完全随机的，并生成 w 位哈希值。由于我们要对所有元素进行哈希，根据鸽巢原理，我们要求 $w = \Omega(\log n)$，以便区分所有元素。

HLL 草图 [FFGM07] 由一个包含 m 个寄存器的数组 M 组成，每个寄存器都记录映射到该寄存器的子流哈希值的最大秩 。所有寄存器都初始化为零。收到新元素 y后，将该元素哈希为一个 w位的值 $x = h(y)$。哈希值分为两部分：用 x 的 $\log_2 m$ 位来选择寄存器索引 $j \in [m]$，其余位用于计算秩 $r = \rho(x)$，即哈希值剩余部分中最左侧 1 位的位置（从 1 开始计数）。显然， $r \sim$ 服从几何分布(1/2)。因此，更新规则为 $M[j] \leftarrow \max\{M[j], r\}$。该草图显然具有幂等性，并且可以通过对寄存器逐元素取最大值轻松合并。

HLL 草图的大小为 $O(m \log\log n)$ 位，因为每个寄存器记录 w位哈希值的一个索引，该索引可用 $O(\log w) = O(\log\log n)$ 位存储。假设哈希值可以在常数时间内计算，更新时间为 $O(1)$。除草图外，还需要 $w = O(\log n)$ 个辅助位来存储哈希值 x，因此草图的总大小为 $O(m \log\log n + \log n)$ 位。我们略去估计算法，因为它与本文无关。

2.4 霍夫曼编码

对具有概率 p_r 的 n 个符号，霍夫曼编码 [Huf52] 的构造方法是：将每个符号 r指定为一棵权重为 p_r的单节点树，然后递归地合并权重最小的两棵树，用一个记录这两棵树总权重之和的节点将它们连接起来，直到只剩下一棵树。若 $p_1 \le p_2 \le \cdots \le p_n$，即概率已排序，则该过程可在 $O(n)$时间内完成 [vL76]。所得的树是一棵具有 $n-1$ 个内部节点和 n 个叶节点的满二叉树；码字通过从根节点遍历到与该符号对应的叶节点获得，将0和1分别赋给左右边，并拼接这些位。令 L 为码字长度的随机变量。众所周知，期望码字长度至多为该分布的熵加一位，即$\mathrm{E}[L] \le H(R) + 1 = 1 - \sum_{r=1}^{n} p_r \log_2 p_r$。

3 算法和数据结构

本节将介绍该算法及其相关数据结构。在某些地方，我们会提前引用第4节中证明的命题，但算法和数据结构的描述是自洽的，不需要了解相关分析。本节的描述旨在进行渐近分析，而实际实现可能会在某些方面有所不同，以改进主要常数；这些内容将在第6节中讨论。

[2]标准教材参考资料请参见 [MU05]。

3.1 哈夫曼桶草图数据结构

哈夫曼桶草图数据结构由以下部分组成

- 一个由 $\frac{m}{B}$ 个桶组成的数组，每个桶包含 B 个寄存器，
- 一棵全局哈夫曼树，或一个将秩映射到变长码字的码本，
- 一个全局基数估计 $\hat{n}$，以及
- 上次构造哈夫曼树时的基数估计 $\hat{n}_{\text{old}}$。

每个桶依次由以下部分组成

- 一个以变宽哈夫曼码字编码的 B 寄存器数组 s，
- 一个包含寄存器码字长度的一元编码的数组，
- 桶内寄存器的桶内最小秩 $r_{\min}$，
- 最小秩寄存器的数量 $c_{\min}$，以及
- 桶内基数估计 $\hat{n}_b$。

该数据结构只需将所有桶 b 的估计值初始化为 $\hat{n}_b = \hat{n} = 0$，使用一棵简单的霍夫曼树，[3] 将所有寄存器值编码为对应秩 0 的码字，并为所有桶设置 $r_{\min} = 0$ 和 $c_{\min} = B$。

设 $B = O(\log n)$。假设码字长度为 $l_1, l_2, \ldots, l_B$，并令 $L = \sum_{j=1}^{B} l_j$ 为该桶中码字的总长度。我们将在命题 6 中证明，码字数组以高概率至多有 $L = O(\log n)$ 位。对 l_j 的一元编码为 $1^{\ell_j}0$，因此一元数组占用 $\sum_{j=1}^{B}(l_j + 1) = O(\log n)$ 位。最小秩以及最小秩寄存器的计数需要 $O(\log \log n)$ 位，基数估计需要 $O(\log n)$ 位。因此，一个桶的总大小为 $O(\log n)$ 位。进而，桶数组的总大小为 $\frac{m}{B} \cdot O(\log n) = O(m)$ 位。

–

根据命题 3，各桶的寄存器与整体草图具有相同的分布，因此我们只需为整个草图维护一棵全局哈夫曼树，它使用全局基数估计 $\hat{n}$ 构建。此外，根据命题 5，全局基数估计只是对潜在基数参数的更好估计，而该参数在固定 m 时明确决定桶中寄存器值的分布，从而决定霍夫曼树。估计 $\hat{n}$ 代表我们对 n 的最佳认知。当某个桶的基数非常小时，最小秩以及最小秩寄存器的计数使我们能够简单地回退到线性计数，作为小基数校正，如同在 [FFGM07] 中那样。

霍夫曼树是一棵包含 $O(\log n)$ 个节点的满二叉树，因此可以用 $O(\log n)$ 位来编码，存储树的结构，并将每个节点映射为叶节点或内部节点。全局基数估计的大小为 $O(\log n)$ 位。因此，我们可以给出以下定理。

定理 1. 哈夫曼桶草图数据结构的总大小为 $O(m + \log n)$ 位，这是最优的 *[AMS99、IW03、KNW10、Woo04]*。

证明。 这直接由数据结构的定义和命题 6 得出。 □

若假设 $m = \Omega(\log^2 n)$，我们可以改善草图上操作的运行时间。这样就可以用两个查找表替代霍夫曼树：一个将秩映射为码字的数组，占用 $O(\log^2 n)$ 位；以及一个将码字映射为秩的哈希表，同样占用 $O(\log^2 n)$ 位。这样一来，草图的总大小便为 $O(m)$ 位。

3.2 哈夫曼桶草图的操作

该草图支持四种操作：窥探、Poke、插入和合并。窥探操作接收一个按桶索引对 $(b,\ j)$，并返回桶 b 中 j^{th} 寄存器的值。Poke 操作接收秩值 r 和一个按桶索引对 $(b,\ j)$，并将桶 b 中的 j^{th} 寄存器设为 r。插入操作接收一个元素 $y \in \mathcal{U}$，并使用 y 更新草图。合并操作接收两个草图，并将它们合并为一个草图。这些操作的伪代码见附录 G。

[3] 当 n = 0，由于 $\Pr[R_j > 0] = 0$，霍夫曼树并未被明确定义，但仍应用哈夫曼算法会得到 s a 简单的梳状树，对应于 r 的一元编码。

3.2.1 Peek（查找寄存器值）

寄存器通过按桶索引对（$b,\ j$）$\in [m/B] \times [B] \simeq [m]$ 进行索引。若要获取寄存器的值，且不对 n 和 m 作任何假设，我们需要在数组中定位码字，并使用霍夫曼树对其解码。使用一元数组定位码字的起始位置和长度需要 $O(B) = O(\log n)$ 时间；解码码字需要 $O(\log\ n)$ 时间，因为最坏情况下可能需要遍历整棵霍夫曼树。因此，最坏情况下，窥探操作可能需要 $O(\log n)$ 时间。

如果我们假设 $m = \Omega(\log^2 n)$，解码就会变成常数时间查找。此外，如果我们假设机器的字长为 $\Omega(\log n)$，就能在常数时间内找到码字及其长度，因此窥探操作的运行时间为 $O(1)$。

3.2.2 Poke（设置寄存器值）

给定一个要设置到桶 b 的 j^{th} 寄存器中的新秩值 r，我们需要将 r 编码为一个码字，并用新码字替换旧码字。如果新旧码字长度不同，可能需要对桶的右侧进行移位，以便容纳新码字。在不对 n 和 m 作任何假设的情况下，编码和移位需要 $O(\log n)$ 时间。然而，如果我们假设 $m = \Omega(\log^2 n)$，并且能够在常数时间内修改 $\Omega(\log n)$ 位，那么编码和移位操作的时间就会变为 $O(1)$。

3.2.3 插入（使用新元素更新草图）

给定元素 $y \in \mathcal{U}$，使用三个不同的哈希函数 $h_b:\ \mathcal{U} \to [m/B]$、$h_j:\ \mathcal{U} \to [B]$ 和 $h_r:\ \mathcal{U} \to [2^w]$（或者使用一个哈希函数并选择适当数量的独立位）来计算桶索引 $b = h_b(y) \in [m/B]$、寄存器索引 $j = h_j(y) \in [B]$ 以及秩 $r = \rho(h_r(y)) \in [w]$。

如果 $r \le r_{\min}$ 对于桶 b 成立，则更新变为空操作。否则，我们需要 Peek 旧的寄存器值 r_{old}。如果 $r_{\text{old}} \ge r$，则再次变为空操作。只有在 $r_{\text{old}} < r$ 时才会触发实际更新。更新通过使用新的秩 r 对寄存器（b, j）执行 Poke 来完成。Poke 之后，如果 $r_{\text{old}} = r_{\min}$，我们需要设置 $c_{\min} \leftarrow c_{\min} - 1$；如果 $c_{\min}$ 变为零，则需要通过 Peek 桶中的所有寄存器来重新计算 $r_{\min}$ 和 $c_{\min}$。我们还需要更新局部基数估计 $\hat{}n_b$ 和全局基数估计 $\hat{}n$。如果全局估计值相较于 $\hat{}n_{\text{old}}$ 变化过大，则需要重构霍夫曼树，并对所有桶重新编码。

在不对 n 和 m作任何假设的情况下，窥探与修改操作需要 $O(\log n)$ 时间，更新 $r_{\min}$ 需要 $O(\log^2 n)$ 时间，而更新估计值需要 $O(\log n)$ 时间。根据分布的单峰性（见第4.2节），可以在 $O(1)$ 时间内轻松确定唯一众数，在 $O(\log n)$ 时间内对概率排序，并在 $O(\log\ n)$ 时间内构造霍夫曼树。对所有桶重新编码需要 $O(m \log n)$ 时间。因此，最坏情况下的运行时间为 $O(m \log n + \log^2 n)$。我们稍后将在定理 8 中证明，假设 $m = O(n/\log^3 n)$，则整个数据流上的每次插入的摊销运行时间为 $O(1)$。m 上界的直观原因是：如果 m 过大，草图就会变得过于稀疏，因为几乎所有寄存器都为零；在这种情况下，使用线性计数反而更合适 [WZT90]。

如果我们假设 $m = \Omega(\log^2 n)$，并且可以在常数时间内修改 $\Omega(\log n)$ 位，那么 Peek、Poke 和估计更新[4] 变为 $O(1)$，而更新 $r_{\min}$ 变为 $O(\log n)$。重构霍夫曼树仍为 $O(\log n)$，但码字的重新编码仅为 $O(m)$。因此最好情况为 $O(1)$，最坏情况为 $O(m)$，其中哈夫曼树构造包含了对所有桶的重新编码。在此情形下，如定理 9 所示，我们在整个数据流上每次插入的摊销运行时间也为 $O(1)$，但我们可以轻易放宽对 m 的假设，改为假设 $m = O(n/\log^2 n)$。因此，该算法在 m 充分大于哈希值、但又显著小于 n 以避免稀疏性的情形下表现最佳，这对应于现实世界的情形。

3.2.4 合并（将两个草图合并为一个草图）

可合并性直接继承自 HLL，因为这些草图是 HLL 草图的无损压缩。执行合并操作时，我们必须解码所有寄存器值，但有可能复用霍夫曼树；如果新的估计值没有大到一定程度，也可能不必对所有值重新编码。最理想的情况是，其中一个草图的全局估计值远大于另一个，此时较小的草图几乎没有贡献，我们可以复用较大草图的霍夫曼树，并基本上直接复制码字。最糟糕的情况是两个草图的估计值相近，

[4]至少在使用原始 HLL 估计器时是这样，它就是调和平均数乘以一个常数。

但它们代表互不相交的元素集合；此时新的估计值约为原有两个估计值中较大者的两倍，因此必须重构霍夫曼树，并对所有寄存器重新编码。

合并操作分两遍进行。首先，我们需要解码所有寄存器值，以计算新的按桶估计值和全局估计；在不对 n 和 m做假设的情况下，耗时 $O(m \log n)$。然后，如有必要，我们构造新的霍夫曼树，耗时 $O(\log n)$。最后，我们需要用新的霍夫曼树重新编码所有寄存器，耗时 $O(m \log n)$。

如果我们能假设 $m = \Omega(\log^2 n)$，并且能在常数时间内访问 $\Omega(\log n)$ 位，则解码变为常数时间查找，重新估计和重新编码变为 $O(m)$ 时间。因此，在这些假设下，合并操作耗时 $O(m)$，因为哈夫曼树构造被重新编码所有寄存器的 $O(m)$ 时间所涵盖。

定理 2. 合并操作在最坏情况下耗时 $O(m \log n)$。然而，假设 $m = \Omega(\log^2 n)$，并且 $\Omega(\log n)$ 位可以在常数时间内访问，则合并操作可以在 $O(m)$ 时间内实现。

证明。 该结果可直接由上述合并操作的描述得出。 □

4 泊松化球与箱模型分析

现在，我们在标准的球与箱模型中分析该算法。本节将始终采用分布的泊松近似，将球的数量 n 视为随机变量，并假设所有 m 个寄存器相互独立且服从相同分布，平均接收 $\lambda = \frac{n}{m}$ 个球。我们将 λ 称为负载因子。

4.1 秩分布

假设有 m 个寄存器，我们来考察固定寄存器 $j \in [m]$ 在经过 n次插入后的秩分布。考虑一个有 m 行、列数无限的箱矩阵。我们独立地将球投入各个箱中，使每个球以均匀随机的方式投入随机一行 j，并以概率 2^{-r}投入第 $r \geq 1$ 列。在泊松模型中，这意味着箱 (j, r) 中的球数 $N_{j,r}$服从 $N_{j,r} \sim \mathrm{Pois}(\lambda 2^{-r})$ 分布，且对所有 $j \in [m]$ 和 $r \geq 1$均相互独立。若考虑指示变量矩阵 $I_{j,r} = [\![N_{j,r} > 0]\!]$，它对应于 FM85 矩阵 [FM85]，该矩阵是整个 LogLog 算法系列的基础。

因此，寄存器值 $R_j = \max_{r \geq 1}\{r \mid N_{j,r} > 0\}$ 是使得 $N_{j,r} > 0$ 成立的最大 r；若不存在这样的 r，则为零。根据泊松分布的性质，寄存器值具有如下分布：

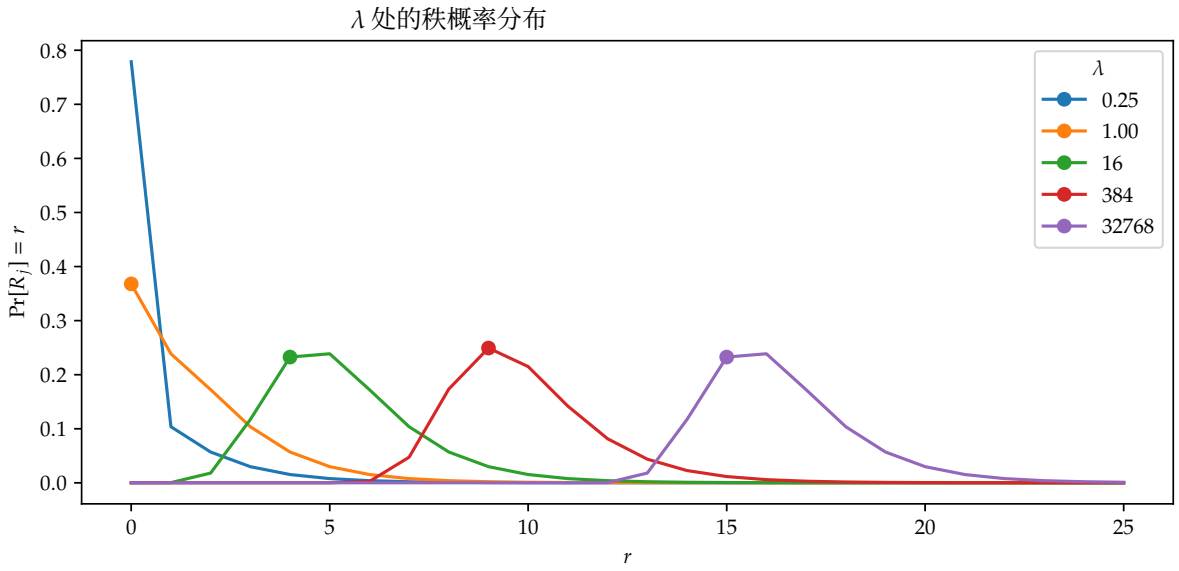
- $\Pr[R_j \leq r] = \prod_{k=r+1}^{\infty} \Pr[N_{j,k} = 0] = e^{-\lambda 2^{-r}}$，对于 $r \geq 1$,
- $\Pr[R_j = 0] = \Pr[R_j \leq 0] = e^{-\lambda}$,
- $\Pr[R_j = r] = \Pr[R_j \leq r] - \Pr[R_j \leq r-1] = e^{-\lambda 2^{-r}} - e^{-\lambda 2^{-(r-1)}} = e^{-\lambda 2^{-r}} - e^{-2\lambda 2^{-r}} = e^{-\lambda 2^{-r}}(1 - e^{-\lambda 2^{-r}})$, 对于$r \geq 1$,
- $\Pr[R_j > r] = 1 - \Pr[R_j \leq r] = 1 - e^{-\lambda 2^{-r}}$，对于 $r \geq 0$，以及
- $\Pr[R_j \geq r] = \Pr[R_j > r-1] = 1 - e^{-\lambda 2^{-(r-1)}} = 1 - e^{-2\lambda 2^{-r}}$ 对于 $r \geq 1$。

由于我们大量使用 $\Pr[R_j = r]$ 的定义，因此当上下文明确 p_r、且 j 无关紧要时，我们将此概率简记为 λ。有时，将概率表示为指数函数也很方便，因此我们记 $x_r = x_r(\lambda) = \lambda 2^{-r}$，以及 $g(x) = e^{-x}(1 - e^{-x})$，从而 $p_r = g(x_r) = e^{-x_r}(1 - e^{-x_r})$。

假设我们将寄存器划分为 m/B 个桶，每个桶包含 B 个寄存器。那么，寄存器值仍具有相同的分布。

命题 3。 考虑这样一个包含 B 个寄存器的桶，其期望接收 $n/(m/B)$ 个球。那么，该桶中寄存器值的分布与包含 n 个球和 m 个寄存器的完整草图相同。

证明。 考虑一个包含 B 个寄存器、期望拥有 $n/(m/B)$ 个球的草图。通过直接计算可得 $\lambda = \frac{n/(m/B)}{B} = \frac{n}{m}$，这与完整草图相符。□



Fig图 1：不同 λ 对应的寄存器值分布，标记表示 $r^* = \lceil \log_2 \lambda \rceil$。

4.2 单峰性与尾部

该分布高度集中在 $r^* = \lceil \log_2 \lambda \rceil$ 附近。直观上，这很合理：每个寄存器本质上是在估计其子流中元素数量的对数，其期望为 $\log_2 \lambda$。秩概率的尾部衰减非常迅速，但并不对称。右尾呈指数衰减，而左尾呈双指数衰减，附录 A 对此进行了说明。事实上，这正是 HLL 的调和平均数比 LogLog 的算术平均数能够提高估计准确度的原因：调和平均数会抑制右尾中的离群值。图 1 展示了 λ 取不同值时的秩分布，同时给出了代表 r^* 的标记。其形状表明该分布是单峰的。

命题 4. 秩分布是单峰的，其众数为 r^*-1、r^* 或 r^*+1，其中 $r^* = \lceil \log_2 \lambda \rceil$。

证明。 参见附录 B。 □

4.3 组合估计

命题 3 已经证明，这些桶的分布与完整草图相同。这意味着，全局基数估计 $\hat{n} = \sum_{b=1}^{B} \hat{n}_b$ 只是对潜在基数 n 更准确的估计。

命题 5. 设 $\hat{n} = \sum_{b=1}^{m/B} \hat{n}_b$ 为按桶估计值之和，其中每个估计值的相对标准误差为 $O(1/\sqrt{B})$。那么，$\hat{n}$ 的相对标准误差为 $O(1/\sqrt{m})$。

证明. 参见附录 C。 □

4.4 桶大小

对于 $j \in [B]$，设 L_j 为桶中 j^{th} 码字的长度。现在我们将证明，每个桶中的码字数组 $L = \sum_{j=1}^{B} L_j$ 的总大小以高概率为 $O(B) = O(\log n)$ 位。思路是注意到：(i) 寄存器值分布的熵为 $O(1)$ 位；(ii) 根据霍夫曼编码的最优性，这意味着码字的期望长度为 $O(1)$ 位；然后 (iii) 利用霍夫曼码字最坏情况下长度的已知上界以及切尔诺夫界，证明以高概率有 $L = O(\log n)$ 位。

命题 6. 设 $B = \alpha \log n$，其中常数为 $\alpha > 0$。则存在常数 $\Gamma, \gamma > 0$，与 n和 m无关，使得对于所有 $t \geq 0$，$\Pr[L \geq \Gamma \log n + t] \leq e^{-\gamma t}$。

证明。 参见附录 D。 □

4.5 霍夫曼树重构次数

现在我们将证明，在 n 次插入过程中，霍夫曼树重构的次数为 $O(\log n)$。这一结果背后的直觉是，根据命题 4，分布是单峰的；而根据附录 A，尾部非常陡峭。因此，分布在尾部非常稳定，不可能在对应尾部的部分发生霍夫曼树变化，因为此时树会塌缩成具有预定义合并顺序的链或“梳状树”结构。此外，由于左尾的衰减速度远快于右尾，我们可以将分布划分为三部分：左尾、众数周围的固定宽度带 以及右尾。左尾必须先塌缩成梳状树，右尾才会对树产生影响；而右尾必须先塌缩成梳状树，带才会对树产生影响。因此，树的所有非平凡变化都必须发生在带中。由于带由众数决定，而众数近似于 n 的以 2 为底的对数，这表明当 λ 越过 2 的幂时，分布会向右移动，并大致发生非平凡变化。这最多只能改变 $O(\log n)$ 次，因此霍夫曼树的变化次数为 $O(\log n)$。

命题 7. 设 $\lambda_n = \frac{n}{m}$对某个固定的 m成立，且 N 是 n的一个上界。树变化的总数 a s λ在 λ上变化 $_1$、λ_2、.....、λ_N是 $O(\log N)$。

证明。 见附录 E。 □

4.6 摊销更新

现在我们来分析更新的摊销成本。更新分为三类：(i) 普通更新，(ii) 最小秩更新，以及 (iii) 树更新。在第4.5节中，我们已将总计 N 次更新中的树更新次数界定为至多 $O(\log N)$，因此接下来只需界定普通更新和最小秩更新的次数。直观来看，只要 λ 足够大，这类更新就不会太多。这是因为草图一旦变得足够“饱和”，几乎没有新元素能触发更新，因为秩分布按几何级数递减。反过来，如果 m 接近 n，草图就会非常稀疏，大多数寄存器都为零，这意味着无论如何我们最好使用线性计数 [WZT90]。

定理 8. 假设一次普通更新需要 $O(\log n)$ 时间，一次最小秩更新需要 $O(\log^2 n)$ 时间，而一次霍夫曼树更新后重新编码所有码字需要 $O(m \log n)$ 时间；此外，假设 $m = O(n/\log^3 n)$，则在 n 次更新中，更新的摊销成本为 $O(1)$。

证明。 见附录 F。 □

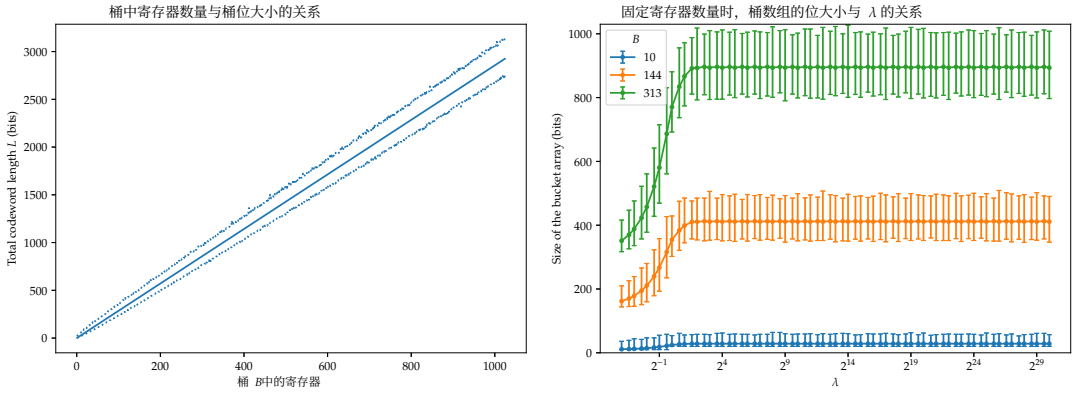
定理 9。 假设我们可以在常数时间内访问 $\Omega(\log n)$ 位，并且 $m = \Omega(\log^2 n)$，从而可以维护一个霍夫曼码本，因此可以假设一次普通更新耗时 $O(1)$，一次最小秩更新耗时 $O(\log n)$，而一次霍夫曼树更新及随后对所有码字的重新编码耗时 $O(m)$，则若进一步假设 $m = O(n/\log^2 n)$，在 n 次更新上的摊销成本为 $O(1)$。

证明。 见附录 F。 □

5 推广

到目前为止，我们将 HBS 草图的分析专门集中在 HLL 寄存器的无损压缩上。然而，该框架允许各种推广，我们在此仅作简要概述，而不对其进行详细分析。

最直接的推广或许是改变秩或 ρ 函数的基数。我们不再查看哈希值的二进制表示中第一个 1 位的位置，而是查看哈希值在任意基数为 b 的表示中第一个非零数字的位置。这与 [Ert24、PW21] 中所述的方向颇为相似。将这种替代秩函数结合进 HBS 框架也可以直接实现，因为压缩过程与被压缩的具体符号无关，



(a) (b)

图 2: （a）桶数组的位大小或码字总大小 L 与桶中寄存器数量 B的关系。实线表示一百万次重复实验的均值，虚线表示最小和最大大小。（b）固定寄存器数量 B 的桶大小与负载因子 λ 的关系，统计基于一百万个随机草图。标记表示均值，误差线表示最小和最大大小。

而且应该也能对桶的行为获得类似的保证。秩函数甚至可以完全不同，只要它具有类似的尾部行为即可。

其次，当前分析基于霍夫曼编码，但除了它简单、易于实现且易于分析之外，并没有理论依据要求我们必须采用它。我们可以考虑其他方案，例如算术编码，前提是它们也能得到同样高效的更新，并且可以证明码字不需要过于频繁地更新。

第三，我们不必局限于 HLL 草图，也可以考虑将同一框架应用于其他草图，利用 FM85 矩阵 [PW21], 中包含的额外信息，例如 ULL [Ert24] 或 ExaLogLog [Ert25]。

最后，该框架也应能应用于其他问题，例如压缩 Count-Min 草图 [CM05], 。它通过哈希函数维护一个整数计数器数组，用于估计数据流中各个元素的频率。Count-Min 草图的关键难点在于，必须为计数器值的分布建立某种模型，因为我们无法像处理 HLL 那样抽象掉数据依赖性。不过，在对数据作出温和假设的情况下，或许可以将 CMS 的行视为桶并进行压缩；也可以在学习数据的同时学习分布。不过，这个问题暂且留作开放问题。

6 实际意义

6.1 实现方面的考量

设计 HBS 草图时考虑了实际实现。采用分桶的想法源于这样一个考虑：桶应足够小，以便能压缩到单个机器字或单条缓存行中。第4节的分析对于实际实现的参考价值有限，因为现实并非渐近情形，常数也很重要。因此，例如，我们实际上无法使用数组中的一元索引这一想法。不过，对于所有实际应用中的 n 值，$\log n$ 都是一个常数，因为实践中使用的是 64 位哈希函数。这使得访问时间为常数，不会真正受寄存器数量或基数的影响。如果要进一步缩短访问时间，可以为 $\lceil \log_2 B \rceil - 1$ 个等间距寄存器的位位置提供桶内索引，从而将解码 $\Theta(B)$ 个寄存器的需求降至 $\Theta(B/\log B)$ 个寄存器；这在实践中有显著提升，因为 B 通常很小。同时不难看出，草图的大小主要由各桶的码字决定，因此，举例来说，霍夫曼编码的码本与之相比微不足道。

图 图 2a 展示了桶的总码字长度 L 随桶中寄存器数量变化的情况 of

表1：在 $m = 2^{15}$ 下，不同位预算时草图的内存使用量和内存-方差乘积（MVP）。

位预算	B	小型草图总大小	大型草图总大小	MVP（小型）	MVP（大型）
64 位	10	242679 位	310800 位	7.961	10.196
512 位	144	119657 位	143111 位	3.926	4.695
1024位	313	108311位	125784位	3.553	4.127

。对于每个 B，通过绘制一百万个随机草图计算这些数值，其中 $n = 2^{30}$、 $m = 2^{15}$，即 $\lambda = 2^{15}$，并使用霍夫曼编码对桶进行编码。实线表示平均值，点线表示最小值和最大值。正如第4节的分析所示，大小随 B线性增长，最小值和最大值也一样；这意味着码字数组的大小由桶中的寄存器数量决定。分析还表明，由于分布形状如图 1 所示，当编码由峰值约在 $r^* = \lceil \log_2 \lambda \rceil$ 附近的常宽带决定时，假设 λ 足够大，使左尾能够显现，桶的位大小与 λ无关。如果左尾没有显现，桶的大小只会更小。因此，当 $\lambda \to \infty$时，桶的大小渐近为常数，仅取决于B。图 2b 展示了在三种不同的 B取值下， λ 取不同值时的这一表现。标记表示平均值，误差棒表示最小值和最大值。

因此，实践中的做法是为码字数组选择一个位预算 b_B ，然后选择 B ，使得有足够的余量，从而任何桶都不太可能溢出。我们在此针对 $n = 2^{30}$ 和 $m = 2^{15}$ 的任意情形给出一个初步分析，其取值来自图2a的随机草图，并不试图严格优化参数，而只是为了了解实际影响。我们考虑三种这样的预算：64位、512位和1024位，分别对应一个机器字、一个缓存行和两个缓存行。如果我们根据图2a的数据选择合适的 B 值，则分别得到10、144和313。

除了每个桶的这个数组之外，我们还需要为每个桶存储最小秩 $r_{\min}$ 以及最小秩寄存器的计数 $c_{\min}$ ，它们分别占用6位和 $\lceil \log_2 B \rceil$ 位。我们还需要存储全局估计$\hat{}n$ 和$\hat{}n_{\text{old}}$，以及霍夫曼树。与理论描述不同，我们不想存储局部估计，因为那需要64位，开销过大；但如果我们对全局$\hat{}n$使用标准估计器 [FFGM07] ，这就没有问题，因为它只是调和平均数，可以在常数时间内轻松更新。如果我们只存储叶节点和内部节点的指示位，霍夫曼树仅占用127位。由于有 $\lceil \frac{m}{B} \rceil$ 个桶，草图的总大小为$\lceil \frac{m}{B} \rceil \cdot (b_B + 6 + \lceil \log_2 B \rceil) + 127$ 位。总大小显示在表1的小草图 列中。显然，总大小随位预算的增加而减小，这体现了草图大小与更新时间之间的权衡。

为了实现更精简的更新，我们可以在一个桶内存储指向对数个寄存器的索引，每个桶需要额外的 $(\lceil \log_2 B \rceil - 1) \cdot \lceil \log_2 n_B \rceil$ 位，并将霍夫曼编码存储为一对查找表。编码表需要$(63 + 6) \cdot 64 = 4416$ 位，因为我们需要为每个可能的值 r存储码字及其长度；解码表则是一个大小至少为$(63 + 6 + 6) \cdot 64 = 4800$ 位的哈希表，因为我们需要将码字及其长度作为键存储，并将值 r 作为值存储。草图的总大小见表1中大草图 一列。

最后需要指出的是，实践中还可以进一步缩小草图的大小，因为我们在这里使用了大量余量，而真正需要我们所使用的位预算的桶非常少。直观地说，当某个（通常是单个）出乎意料的大秩值落入某个桶时，就会发生溢出，从而需要很长的码字。或许可以采用自适应方案：大多数桶用更少的位存储，同时分配少量尺寸更大的大桶 ，以容纳少数发生溢出的桶。

6.2 内存-方差乘积

HLL 等草图的性能通常用内存-方差乘积（MVP） [PW21], 来衡量，它是内存用量与估计量的相对方差 $\text{Var}\left[\frac{\hat{n}}{n}\right]$的乘积。第4节的分析是渐近分析，忽略了常数，因此除了表明 MVP 必然为 $O(1)$ 之外，并不能直接给出有意义的结果。不过，我们可以通过对其行为进行数值建模，初步证明该算法在这些指标下具有实用性。A

正如我们所 t analyzing the estimators in thi本文，我们 hall 假设 the relati有方差iance is bounded

为 1.075/m [WP23]。正如我们在第6.1节中观察到的，桶的大小仅由 B 的选择决定，与 λ无关；而草图的大小主要由码字数组决定，因此渐近而言，草图大小为 $O(m)$。固定桶的位预算后，我们便可以通过数值计算得出常数。这样就能计算不同位预算下的 MVP，如表1所示。与 ExaLogLog [Ert25], 相比，即便采用这里介绍的简单方案，且未充分优化参数和实现细节，我们也能在不使用 FM85 矩阵额外信息的情况下，使 MVP 达到与 ExaLogLog 的 3.67 相当的水平；此外，我们还可以将该框架应用于 [Ert25] 的框架。

需要指出的是，这并非严谨或完整的分析，而是一个粗略估算，旨在说明从内存-方差乘积（MVP）的角度来看，该草图具有实用性。精确内存用量和内存-方差乘积的更严谨分析留待未来工作。

致谢

我要感谢 Rasmus Pagh 和 Martin Aumüller 提供的有益讨论和意见。我还要感谢 Tamás Sarlós 提出的好点子，以及那个精彩的比喻：像 Baron von Münchhausen 一样，靠着拽自己的头发把自己从沼泽里拉出来。最后，我要感谢我的硕士生 Labiba Karar Eshaba 和 Yasamin Fazelidehkordi，因为本文中的一些构想源于我们围绕她们关于一种相关但不同的基数估计方法的硕士论文项目所进行的讨论。

参考文献

[AMM00] Yaser S. Abu-Mostafa 和 Robert J. McEliece。霍夫曼编码中的最大码字长度。计算机与数学应用，39(11):129–134，2000。[AMS99] Noga Alon、Yossi Matias 和 Mario Szegedy。近似频率矩的空间复杂度。计算机与系统科学杂志，58(1):137–147，1999。[BEF+18] Ran Ben-Basat、Gil Einziger、Shir Landau Feibish、Jalil Moraney 和 Danny Raz。面向网络的路由无关重频元素检测。载于*2018*年网络与通信系统架构研讨会论文集（*ANCS 2018*），第66–73页，2018年。[BL19] Daniel N. Baker 和 Ben Langmead。Dashing：借助 HyperLogLog 快速、准确地计算基因组距离。基因组生物学，20(1):265，2019。[Bla20] Jaroslaw Blasiok。以高概率实现不同元素的最优流式处理与跟踪。*ACM*算法汇刊，16(1):3:1–3:28，2020。[CCD14] Yousra Chabchoub、Raja Chiky 和 Betul Dogan。滑动HyperLogLog与EWMA如何检测IP流量中的端口扫描攻击？*EURASIP*信息安全期刊，2014:5，2014。[CDP+19] Pern Hui Chia、Damien Desfontaines、Irippuge Milinda Perera、Daniel Simmons-Marengo、Chao Li、Wei-Yen Day、Qiushi Wang 和 Miguel Guevara。KHyperLogLog：大规模估计大型数据的可再识别性与可连接性。载于*2019*年*IEEE*安全与隐私研讨会论文集（*SP 2019*），第350–364页，2019年。[CM05] Graham Cormode 和 S. Muthukrishnan。改进的数据流摘要：Count-Min 草图及其应用。算法杂志，55(1):58–75，2005。[CSGH23] Vera Clemens、Lars-Christian Schulz、Marten Gartner 和 David Hausheer。利用HYPERLOGLOG和COUNTMIN草图在P4中进行DDoS检测。载于*IEEE/IFIP*网络运营与管理研讨会论文集（*NOMS 2023*），第1–6页，2023年。[CT05] Thomas M. Cover 和 Joy A. Thomas。信息论基础。John Wiley & Sons, Inc.，美国新泽西州霍博肯，第二版，2005年。[DF03] Marianne Durand 和 Philippe Flajolet。大基数的LogLog计数（扩展摘要）。载于第*11*届欧洲算法年会论文集（*ESA 2003*），计算机科学讲义第2832卷，第605–617页，2003年。

[DHS+23] Yang Du、He Huang、Yu-e Sun、Kejian Li、Boyu Zhang 和 Guoju Gao。一种位数更少、更新时间为常数且可合并的改进基数估计器。载于IEEE计算机通信会议（INFOCOM 2023）论文集，第1–10页，2023年。[DJJ+22] Dinhnguyen Dao、Rhongho Jang、Changhun Jung、David Mohaisen 和 DaeHun Nyang。降低基于HyperLogLog的多流扩散估计中的噪声。载于第52届IEEE/IFIP年度可靠系统与网络国际会议（DSN 2022）论文集，第331–342页，2022年。[Dur04] Marianne Durand。数据集的解析组合学与算法学。（多变量全纯性、组合学应用与算法分析）。博士论文，巴黎综合理工学院，法国帕莱索，2004年。[Ert17] Otmar Ertl。HyperLogLog草图的新基数估计算法。CoRR，abs/1702.01284，2017年。[Ert24] Otmar Ertl。UltraLogLog：用于近似去重计数的实用且更节省空间的HyperLogLog替代方案。VLDB基金会会刊，17(7):1655–1668，2024年。[Ert25] Otmar Ertl。ExaLogLog：空间高效且实用的艾字节规模近似去重计数。载于第28届数据库技术扩展国际会议论文集（EDBT 2025），第829–841页，2025年。[FFGM07] Philippe Flajolet、Éric Fusy、Olivier Gandouet 和 Frédéric Meunier。HyperLogLog：近最优基数估计算法的分析。载于AofA 2007论文集，2007年。[FM85] Philippe Flajolet 和 G. Nigel Martin。数据库应用的概率计数算法。计算机与系统科学杂志，31(2):182–209，1985年。[FN19] Michael J. Freitag 和 Thomas Neumann。每一行都计数：结合草图与采样，准确估计分组结果。载于第9届创新数据系统研究双年会论文集（CIDR 2019），2019年。[HNH13] Stefan Heule、Marc Nunkesser 和 Alexander Hall。实践中的HyperLogLog：先进基数估计算法的算法工程。载于2013年联合EDBT/ICDT会议论文集（EDBT 2013），第683–692页，2013年。[Huf52] David A. Huffman。一种构造最小冗余编码的方法。IRE论文集，40(9):1098–1101，1952年。[IW03] Piotr Indyk 和 David P. Woodruff。不同元素问题的紧下界。载于第44届计算机科学基础研讨会论文集（FOCS 2003），第283–288页，2003年。[KNW10] Daniel M. Kane、Jelani Nelson 和 David P. Woodruff。一种解决不同元素问题的最优算法。载于第二十九届ACM SIGMOD-SIGACT-SIGART数据库系统原理研讨会论文集（PODS 2010），第41–52页，2010年。[KP22] Matti Karppa 和 Rasmus Pagh。HyperLogLogLog：多一个对数的基数估计。载于第28届ACM知识发现与数据挖掘会议论文集（KDD 2022），第753–761页，2022年。[Lan17] Kevin J. Lang。回到未来：一种更接近最优的基数估计算法。CoRR，abs/1708.06839，2017年。[MSPK19] Guillaume Marçais、Brad Solomon、Rob Patro 和 Carl Kingsford。基因组学中的草图与次线性数据结构。《生物医学数据科学年度评论》，2:93–118，2019年。[MU05] Michael Mitzenmacher 和 Eli Upfal。《概率与计算：随机化算法与概率分析》。剑桥大学出版社，纽约，2005年。[PCT23] Christina Pavlopoulou、Michael J. Carey 和 Vassilis J. Tsotras。重新审视大数据管理系统中连接查询的运行时动态优化。SIGMOD Rec.，52(1):104–113，2023年。

[PW21] Seth Pettie 和 Dingyu Wang。基数估计的信息论极限：Fisher 遇见香农。载于第*53*届*ACM SIGACT*计算理论年会（*STOC 2021*）论文集，第556–569页，2021年。[PWY21] Seth Pettie、Dingyu Wang 和 Longhui Yin。用于基数估计的不可合并草图。载于第*48*届自动机、语言与编程国际研讨会（*ICALP 2021*）论文集，第104:1–104:20页，2021年。[Tru19] Immanuel Trummer。具有有界执行成本的精确基数查询优化。载于*2019*年数据管理国际会议（*SIGMOD 2019*）论文集，第2–17页，2019年。[vL76] Jan van Leeuwen。关于霍夫曼树的构造。载于第三届自动机、语言与编程国际研讨会（*ICALP 1976*）论文集，第382–410页，1976年。[WCLL21] Jiayi Wang、Chengliang Chai、Jia bin Liu 和 Guoliang Li。FACE：一种基于归一化流的基数估计器。*VLDB*基金会会刊，15(1):72–84，2021年。[Woo04] David P. Woodruff。所有频率矩的最优空间下界。载于第*15*届*ACM-SIAM*离散算法年会（*SODA 2004*）论文集，第167–175页，2004年。[WP23] Dingyu Wang 和 Seth Pettie。适用于 HyperLogLog、PCSA 及其他方法的改进基数估计器。载于第*42*届*ACM SIGMOD-SIGACT-SIGAI*数据库系统原理研讨会（*PODS 2023*）论文集，第317–327页，2023年。[WZT90] Kyu-Young Whang、Brad T. Vander Zanden 和 Howard M. Taylor。一种用于数据库应用的线性时间概率计数算法。*ACM*数据库系统汇刊，15(2):208–229，1990年。[XCZL20] Qingjun Xiao、Shigang Chen、You Zhou 和 Junzhou Luo。通过提高内存效率，对任意大的数据流进行基数估计。*IEEE/ACM*网络汇刊，28(2):433–446，2020年。

附录

A 分布的尾界

我们将证明该分布尾部的以下界限，但不声称这些结果是原创的。

引理10. 设 $r^* = \lceil \log_2 \lambda \rceil$。对于任意整数 $k \geq 0$，我们有 $\Pr[R_j > r^* + k] \leq 2^{-k}$。

证明。 设 $\theta = \lambda 2^{-r^*} \leq 1$。因此，$\Pr[R_j > r^* + k] = 1 - e^{-\theta 2^{-k}} \leq \theta 2^{-k} \leq 2^{-k}$。 □

引理 11. 假设 $\lambda > 1$，并设 $r^* = \lceil \log_2 \lambda \rceil$。对于任意整数 $k \geq 0$，我们有 $\Pr[R_j \leq r^* - k] \leq e^{-2^{k-1}}$。

证明。设 $\theta = \lambda 2^{-r^*}$ $\theta 2^k \geq 2^{k-1}$ $\lambda > 1$ $r^* \geq 1$ $[R_j \leq r^* - k] = e^{-\theta 2^k} \leq$。我们有，因为以及。 □
因此，Pr。$e^{-2^{k-1}}$

B 命题4的证明

我们先考虑该分布的连续类比，将 r 替换为连续变量 x。定义 g： $\mathbb{R} \to \mathbb{R}$ 使得 $g(x) = e^{-x}(1 - e^{-x})$。

引理 12. 函数 g 是单峰的，在 $x = \ln 2$处取得唯一最大值。

证明。 我们有 $g'(x) = e^{-2x}(2 - e^x)$。唯一的临界点位于 $x = \ln 2$ 处，且为极大值，因为当 $x < \ln 2$ 时，$g'(x) > 0$ ；当 $x > \ln 2$ 时， $g'(x) < 0$。□

我们来考虑众数位于 $r = 0$ 且整个序列严格递减的特殊情况。

引理 13. 如果 $x_1 = \frac{\lambda}{2} \leq \ln 2$，那么该分布的众数位于 $r = 0$，且整个序列 p_r 严格递减。

证明。 考虑 $p_0 \geq p_1 > p_2$ 的情况，也就是说，众数位于 $r = 0$，或同时位于 $r = 0$ 和 $r = 1$。只有当 $x_1 = \frac{\lambda}{2} \leq \ln 2$ 时，这种情况才会发生，也就是说， $\lambda \leq 2 \ln 2$。由于 $\lambda \leq 2 \ln 2$，我们有 $p_0 = e^{-\lambda} \geq e^{-2 \ln 2} = \frac{1}{4}$，并且 $p_1 = e^{-\lambda/2}(1 - e^{-\lambda/2}) \leq \frac{1}{4}$，因此这一条件保证众数位于 $r = 0$，且该序列在 p_1 之后严格递减。□

因此我们可以 现在假设 $p_1 > p_0$ ，且众数位于 $r \geq 1$。我们来回顾一下命题
的 4，并对其进行证明。

命题 n 4. 秩分布是单峰的，众数位于 $r^* - 1$、 r^* 或 $r^* + 1$，其中 $r^* = \lceil \log_2 \lambda \rceil$.

证明。 令 r' 为满足 $x_{r'} \leq \ln 2$ 的最小整数，其中 g 取最大值。这就是“交叉指标”，即分布从递增转为递减的位置。根据 g 的性质，当 $r < r'$ 时， p_r 严格递增；当 $r > r'$ 时， p_r 严格递减。若 $x_r = \ln 2$，则可能出现一种特殊情况：此时序列会一直递减到 x_r。不过无论如何，当 $r \geq r'$ 时，序列都是非递增的。因此，众数位于 r' 或 $r' - 1$，取决于哪一个能使 x_r 更接近 $\ln 2$。

根据 $r^* = \lceil \log_2 \lambda \rceil$ 的定义，我们有 $x_{r^*} = \lambda 2^{-r^*} \leq 1$。此外， $x_{r^*-1} = \lambda 2^{-(r^*-1)} = 2 \frac{\lambda}{2^{r^*}}$，且 — $x_{r^*+1} = \lambda 2^{-(r^*+1)} = \frac{1}{2} \frac{\lambda}{2^{r^*}}$。因此， $\frac{1}{2} < x_{r^*} \leq 1$、 $1 < x_{r^*-1} \leq 2$ 和 $\frac{1}{4} < x_{r^*+1} \leq \frac{1}{2}$。此外，$\frac{1}{2} < \ln 2 < 1$，所以交叉指标必须满足 $r^* \leq r' \leq r^* + 1$。众数必须满足以下条件：它位于 r' 或 $r' - 1$。综合以上结果，众数位于 $r^* - 1$、 r^* 或 $r^* + 1$。至此证明完毕，众数为零的特殊情况由引理 13 处理。□

C 命题 5 的证明

让我们回顾一下命题 5 的陈述并给出证明。

命题 5。 设 $\hat{n} = \sum_{b=1}^{m/B} \hat{}n_b$ 为按桶估计值之和，每个估计值的相对标准误差为 $O(1/\sqrt{B})$。则 $\hat{}n$ 的相对标准误差为 $O(1/\sqrt{m})$。

证明。 我们已在命题 3 中证明，一个桶中寄存器值的分布与完整草图中寄存器值的分布相同。这意味着全局基数估计 $\hat{}n = \sum_{b=1}^{B} \hat{}n_b$ 只是对潜在基数 n 的更精确估计。根据原始 HLL 估计器 [FFGM07], 的性质，每个 $\hat{}n_b$ 的相对标准误差以 $O(1/\sqrt{B})$ 为上界。由于桶中预期有 $n/(m/B)$ 个元素，这意味着

$$\frac{\sqrt{\mathrm{Var}[\hat{n}_b]}}{\frac{n}{m/B}} = \frac{\sqrt{\mathrm{Var}[\hat{n}_b]}}{\frac{nB}{m}} = \frac{\sqrt{\mathrm{Var}[\hat{n}_b]}}{n} \cdot \frac{m}{B} \le \frac{K}{\sqrt{B}}$$

对于某个常数 K（根据 [FFGM07], 为 1.04）。对 $\mathrm{Var}[\hat{}n_b]$ 求解可得 $\mathrm{Var}[\hat{n}_b] \le \frac{K^2 n^2 B}{m^2}$。假设按桶估计值相互独立，我们有 $\mathrm{Var}[\hat{n}] = \sum_{b=1}^{m/B} \mathrm{Var}[\hat{n}_b] \le \frac{K^2 n^2 B}{m^2} \cdot \frac{m}{B} = \frac{K^2 n^2}{m}$。因此，$\hat{}n$ 的相对标准误差为 $\frac{\sqrt{\mathrm{Var}[\hat{n}]}}{n} \le \frac{K}{\sqrt{m}}$。 □

D 命题 6 的证明

证明分为两部分。首先，我们证明寄存器值的熵为 $O(1)$。这意味着霍夫曼编码的期望码字长度也为 $O(1)$，据此可以将桶数组的总大小界定为 $O(\log n)$，如所声称的那样。

D.1 熵

我们首先证明寄存器值的熵为 $O(1)$。此处给出完整证明，但我们不声称这一结果是原创的，因为它早已广为人知 [Dur04, PW21]。下面关于 x_r 的界对于界定 $\log p_r$ 很有用。

引理 14。 对于 $r \ge r^* = \lceil \log_2 \lambda \rceil$，$\Pr[R_j = r] = p_r$ 满足 $p_r \ge \frac{x_r}{2e} = \frac{\lambda 2^{-r}}{2e}$。

*Pr*证明。 对于 $r \ge r^*$，有 $e^{-x_r} \ge 1/e$。此外，$1 - e^{-x_r} \ge \frac{x_r}{2}$。综合以上可得 $p_r \ge \frac{x_r}{2e}$. □

现在我们可以证明熵界了。

引理 15. 寄存器值 R_j 的熵为 $H(R_j) = -\sum_{r=0}^{\infty} p_r \log_2 p_r = O(1)$。

证明。 考虑

$$H(R_j) = \sum_{r=0}^{\infty} -p_r \log_2 p_r = (\log_2 e)\left(\sum_{r=0}^{r^*-1} -p_r \ln p_r + \sum_{r=r^*}^{\infty} -p_r \ln p_r\right). \tag{1}$$

我们需要分别界定方程（1）中的两个求和。对于右侧尾部，由于 $1 - e^{-x_r} \le x_r$，我们有 $p_r \le x_r e^{-x_r} \le x_r$。根据引理 14，$p_r \ge \frac{x_r}{2e}$。因此，对于 $r \ge r^*$，我们有

$$\frac{x_r}{2e} \le p_r \le x_r\,. \tag{2}$$

将方程（2）代入方程（1）右侧的求和式，令 $\theta = \lambda 2^{-r^*} \le 1$，则对于 $r = r^* + k$

我们有 $x_r = \theta 2^{-k}$，得到

$$\begin{aligned}
-\sum_{r=r^*}^{\infty} p_r \ln p_r &\le -\sum_{k=0}^{\infty} x_r \ln \frac{x_r}{2e} = -\sum_{k=0}^{\infty} x_r (\ln x_r - \ln 2 - 1) \\
&= -\sum_{k=0}^{\infty} x_r \ln x_r + \sum_{k=0}^{\infty} (\ln 2 + 1) \\
&= -\sum_{k=0}^{\infty} \theta 2^{-k} \ln\left(\theta 2^{-k}\right) + \sum_{k=0}^{\infty} \theta 2^{-k} (\ln 2 + 1) \\
&= -\theta \sum_{k=0}^{\infty} 2^{-k} (\ln \theta - k \ln 2) + \theta(\ln 2 + 1) \sum_{k=0}^{\infty} 2^{-k} \\
&= -2\theta \ln \theta + 2\theta \ln 2 + 2\theta \ln 2 + 2\theta = -2\theta \ln \theta + 4\theta \ln 2 + 2\theta \,.
\end{aligned} \tag{3}$$

由于 $-\theta \ln \theta \le 1/e$ 且 $0 \le \theta \le 1$，方程（3）被 $\frac{2}{e} + 4 \ln 2 + 2 < 6$ 所界定。

对于方程（1）的左侧尾部和左侧求和，当 $r \le r^*$ 时，我们有 $p_r \le e^{-2^{k-1}} \le 1/e$，因此 $-p_r \ln p_r \le 2^{k-1} e^{-2^{k-1}}$。于是，

$$-\sum_{r=0}^{r^*-1} p_r \ln p_r \le \sum_{k=1}^{\infty} 2^{k-1} e^{-2^{k-1}} < 1 \,.$$

综合以上结果，我们有 $H(R_j) < (\log_2 e)(6+1) = O(1)$。 □

注 16。数值上，$H(R_j)$ 很快趋近于 ≈ 2.83196 当 m 固定且 $n \to \infty$ 时。

D.2 桶大小

现在我们将证明，以高概率，每个桶中的码字数组大小为 $O(B) = O(\log n)$ 位。设 L_j 为桶中 j^{th} 寄存器的码字长度这一随机变量，并令 $L = \sum_{j=1}^{B} L_j$ 为桶中码字的总长度。我们需要以下引理来界定 L_j 的最坏情况长度。相应地，令 $L(r)$ 表示对秩值 r 进行编码的码字长度，即 $L_j = L(R_j)$。

引理 17([AMM00], 改述）。假设 L_j 是对概率为 p_r 的符号进行编码的霍夫曼码字的长度。那么，$L_j \le -1.44041$ 对数$_2 p_r$ 是最坏情况。

我们需要以下引理。

引理 18。对于 $r \ge r^*$，我们有 $p_{r^*+k} \ge \frac{2^{-k}}{4e}$。

证明。根据 r^* 的选取，我们有 $\frac{1}{2} < \frac{\lambda}{2^{r^*}} \le 1$，因此对于 $k \ge 0$，我们有 $x_{r^*+k} = \frac{\lambda}{2^{r^*+k}} \ge \frac{1}{2} 2^{-k} = 2^{-(k+1)}$。结合引理 14，我们得到 $p_r \ge \frac{x_{r^*+k}}{2e} = \frac{\lambda 2^{-r^*+k}}{2e} \ge \frac{2^{-(k+1)}}{2e} = \frac{2^{-k}}{4e}$。 □

综合引理 17 和引理 18 可知，对于 $k \ge 0$，

$$L(r^* + k) \le -2 \log_2 p_{r^*+k} \le -2 \log_2 \frac{2^{-k}}{4e} = 2k + 2 \log_2(4e) \,. \tag{4}$$

众所周知（例如，参见 [CT05，第 5 章。7]），霍夫曼编码在期望意义下是最优的，且 $\mathrm{E}[L_j] \le H(R_j) + 1$。因此，根据引理 15 和期望的线性性，我们有 $\mathrm{E}[L] = O(\log n)$。不过，我们还需要证明一个更强的结论：以高概率，$L$ is $O(\log n)$。我们将通过应用标准的切尔诺夫论证来证明这一点。切尔诺夫界如下：设$X_1, X_2, \ldots, X_n$ 是相互独立且同分布的随机变量，它们具有共同的矩生成函数 $M(t) = \mathrm{E}[e^{tX_i}]$，并且该函数对某个 $t > 0$ 有限，且设 $X = \sum_{i=1}^{n} X_i$。那么，对于任意这样的$t > 0$，有

$$\Pr[X \ge a] \le e^{-ta} \prod_{i=1}^{n} M(t) = e^{-ta} \left(\mathrm{E}[e^{tX_i}]\right)^n \,. \tag{5}$$

T要应用方程（5），我们需要证明 L_j 的矩母函数是有限的 te.

Le引理 19. 存在与 n 和 m 无关的常数 $\tau > 0$ 和 $M < \infty$，使得 $\mathrm{E}[e^{\tau L_j}] \le M$. 证明。 固定 $0 < \tau < \ln 2$ 2 。我们有

$$\mathrm{E}[e^{\tau L_j}] = \sum_{r=0}^{\infty} e^{\tau L(r)} p_r = \sum_{r=0}^{r^*-1} e^{\tau L(r)} p_r + \sum_{r=r^*}^{\infty} e^{\tau L(r)} p_r \,. \tag{6}$$

方程（6）的左侧求和是有限的，因此只需证明右侧求和也是有限的。根据方程（4）和引理 18，我们有

$$\sum_{r=r^*}^{\infty} e^{\tau L(r)} p_r = \sum_{k=0}^{\infty} e^{\tau L(r^*+k)} p_{r^*+k} \le \sum_{k=0}^{\infty} e^{\tau(2k+2\log_2(4e))} \cdot \frac{2^{-k}}{4e} = \frac{e^{2\tau \log_2(4e)}}{4e} \sum_{k=0}^{\infty} \left(\frac{e^{2\tau}}{2}\right)^k < \infty \,,$$

因为按照我们对 τ 的选择，有 $\frac{e^{2\tau}}{2} \le 1$。 □

让我们回顾命题 6 的陈述并加以证明。

命题 6。 令 $B = \alpha \log n$，其中 $\alpha > 0$ 为某个常数。则存在常数 Γ、$\gamma > 0$，它们与 n 和 m 无关，使得对所有 $t \ge 0$，$\Pr[L \ge \Gamma \log n + t] \le e^{-\gamma t}$。

证明。 固定 $\tau > 0$ 和 M，使得 $\mathrm{E}[e^{\tau L_j}] \le M$，如引理 19 所保证。根据方程（5）的切尔诺夫界，我们有

$$\Pr[L \ge s] \le e^{-\tau s} \left(\mathrm{E}[e^{\tau L_j}]\right)^B$$

令 $s = \Gamma \log n + t > 0$ 且 $B = \alpha \log n$，于是我们有

$$\Pr[L \ge \Gamma \log n + t] \le e^{-\tau(\Gamma \log n + t)} M^{\alpha \log n} \,.$$

取 $\Gamma \ge \frac{\alpha}{\tau} \ln M + 1$，直接计算可得

$$\Pr[L \ge \Gamma \log n + t] \le e^{-\tau t} \,,$$

由此证明了该结论，其中 $\gamma = \tau$。 □

E 命题7的证明

这是一个冗长且技术性很强的证明，因此拆分成了几个小节。主要思路是证明霍夫曼树的尾部具有很强的刚性，并且会很快变成梳状树，其合并顺序简单且预先确定。这意味着，可能发生有趣树变化的唯一位置，是众数周围一个狭窄的常宽带内。我们首先单独分析右尾，以建立刚性判据。然后，我们定义由大权重构成的常宽带，以及由小权重构成的池。池由深左尾和深右尾组成。我们对左尾和右尾的质量进行界定，并证明整个左尾都被右尾中的每个元素所支配；类似地，右尾的总质量也被带中的每个元素所支配。因此，左尾必须先合并为单个节点，才能与右尾发生相互作用；同样，右尾也必须先合并为单个节点，才能与带发生相互作用。根据罗尔定理，假设 λ 被限制在一个常宽区间内，由于与节点子集对应的概率质量是有限项指数和，这意味着树变化的次数只能是常数。确定了 λ 的一个常数区间内可能存在的交叉点数量为常数后，只需注意到这样的区间只有 $O(\log n)$ 个，即可得到结论。

E.1 基础知识

在本节中，我们假设基数 N 存在一个上界。对于固定的 m，显然，定义霍夫曼树的分布由 n 唯一确定。因此，定义 $\lambda_n = \frac{n}{m}$, $x_r(\lambda) = \frac{\lambda}{2^r}$，并回顾一下，$p_0(\lambda) = e^{-\lambda}$ 和 $p_r(\lambda) = e^{-x_r(\lambda)}(1 - e^{-x_r(\lambda)})$ 对于 $r > 0$ 成立。令 $T(\lambda)$ 表示由分布 $p_r(\lambda)$ 诱导的霍夫曼树。我们将分析变化点 $C_N = \{i \in [N-1] \mid T(\lambda_i) \ne T(\lambda_{i+1})\}$ 的集合，也就是霍夫曼树发生变化处的索引集合。我们的目标是证明 $|C_N| = O(\log N)$。

对于 $\lambda > 0$，令 $r^*(\lambda) = \lceil \log_2 \lambda \rceil$。定义区间 $I_0 = [0, 1]$，并且对于 $k \ge 1$, $I_k = (2^{k-1}, 2^k]$.

引理 20. 如果 $k \geq 1$ 且 $\lambda \in I_k$，则 $r^*(\lambda) = k$ 且 $2^{-k}\lambda \in (1/2, 1]$.

证明。 由 $2^{k-1} < \lambda \leq 2^k$可立即得出。 □

我们还为 $x > 0$定义了 $g(x) = e^{-x}(1 - e^{-x})$。然后，对于 $r \geq 1$， $p_r(\lambda) = g(x_r(\lambda))$，且 $p_0(\lambda) = e^{-\lambda}$。

E.2 单独考察右尾

我们将从单独考察右尾开始分析，也就是说，暂不考虑左尾。我们证明，与右侧尾部对应的霍夫曼树部分具有很强的刚性，会收缩为链或“梳状树”结构，并遵循简单的预定义合并顺序。下面的引理给出了梳状结构出现的判据。

引理 21。 设 $w_0 \geq w_1 \geq \cdots \geq w_M > 0$ 为一个权重序列。若对所有 $i \leq M - 2$，都有

$$w_i \geq w_{i+1} + w_{i+2}\,, \tag{7}$$

then 由 w_i 定义的分布对应的霍夫曼树具有梳状结构，其预定义合并顺 *er.*

证明。 霍夫曼算法在每一步都会合并概率最小的两个符号（或内部节点）。根据式 (7) 中的判据，新形成的和必定小于或等于下一个权重，因此由归纳法可得结论。□

定义 22。 对于 $x > 0$，定义比率函数 $h(x)$如下：

$$h(x) = \frac{g(x/2)}{g(x)} = \frac{e^{-x/2}(1 - e^{-x/2})}{e^{-x}(1 - e^{-x})} = \frac{e^{x/2}}{1 + e^{x/2}}\,.$$

引理 23。 对于 $x > 0$，$h(x)$ 严格递增。

证明。 我们有 $h'(x) = \frac{e^x}{(2+e^{x/2})\,2(1+e^{x/2})}$ □
$_2 > 0$。

令 $q_* = \frac{\sqrt{5}-1}{2} \approx 0.618034$。注意，

$$q_* + q_*^2 = 1\,. \tag{8}$$

选择 $\eta > 0$ 使得

$$\eta \leq 2\ln\left(\sqrt{5} - 1 + \sqrt{6\sqrt{5} - 2}\right) - 4\ln 2 \approx 0.285985\,.$$

重要的是，这保证了 $h(\eta) \leq q_*$。固定整数 $K'_R \geq 1$，使得 $2^{-K'_R} \leq \eta$。现在我们可以证明，分布的右侧尾部在截断点 $r^*(\lambda)+ K'_R$ 之后，由尾部之前的最后一个元素所支配。

引理 24。 设 $k \geq 1$、 $\lambda \in I_k$和 $r \geq k + K'_R$。则， $p_r(\lambda) \geq p_{r+1}(\lambda) + p_{r+2}(\lambda)$。

证明。 根据引理 20，我们 $\lambda 2^{-k} \leq 1$，因此对于 $r \geq k + K'_R$，根据 K'_R 的选取，我们有

$$x_r(\lambda) = \lambda 2^{-r} \leq 2^{-K'_R} \cdot 2^{-k} \cdot \lambda \leq 2^{-K'_R} \leq \eta\,.$$

那么，$\frac{p_{r+1}}{p_r} = \frac{g(x_r/2)}{g(x_r)} = \mathrm{h}(x_r) \leq \mathrm{h}(\eta) = q_*$，并且 $\frac{p_{r+2}}{p_{r+1}} = \frac{g(x_{r+1}/2)}{g(x_{r+1})} = \mathrm{h}(x_{r+1}) \leq \mathrm{h}(\eta) \leq q_*$，因为根据引理 23，h 严格递增且 $x_{r+1} \leq x_r \leq \eta$。因此， $p_{r+2}/p_r = (p_{r+2}/p_{r+1})(p_{r+1}/p_r) \leq q_*^2$，所以根据公式 (8)， $p_{r+1} + p_{r+2} \leq (q_* + q_*^2)p_r = p_r$。□

推论 25. 固定如上所述的 K'_R。对于任意 $k \geq 1$ 和任意 $\lambda \in I_k$，由索引 $k + K'_R$、 $k + K'_R + 1$、. . . 诱导的霍夫曼子树是具有预定义合并顺序的梳状树。

证明。 由引理 21 和 24 可立即得出。 □

E.3 带和池

显然，引理 11 中同样的论证也适用于按双指数速度衰减的左侧尾部，因此，若单独处理，左侧尾部的结构更加僵化，并会按照简单的预定义合并顺序塌缩为梳状结构。然而，由于某些极小的左尾元素也可能与右尾中的某些元素合并，我们需要构造一个包含两侧尾部元素的小元素池。接下来我们将证明，池中的尾部具有僵化结构，在完全塌缩为单个节点之前，不会对树造成任何非平凡的改变。为此，我们截断无限的右尾，以便建立左尾与右尾之间的支配准则：整个左尾都被截断后的右尾中的每个元素支配，因此左尾必须先塌缩为单个节点，才能影响右尾。同样，截断后的整个右尾都被带中的每个元素支配，因此右尾必须先塌缩为单个节点，才能影响带。

暂且假设对于某个 $k \geq 1$ 有 $\lambda \in I_k$，因此该分布的众数为 k，我们可以将 $\lambda = \theta 2^k$ 写成某个 $\theta \in (1/2, 1]$。选择一个与 λ 无关的常数 $0 < \delta < \ln 2$，并确定秩 $r_{\max}$ 的一个整数上界。

引理 26。 固定 $0 < \delta < \ln 2$，使其与 λ 无关，并令 $r_{\max} = \lceil \log_2 \frac{\lambda}{\delta} \rceil$。那么，对于任意 $k \geq 1$ 和 $\lambda \in I_k$，都有$\Pr[R_j > r_{\max}] \leq \delta$。

证明。我们有

$$\Pr[R_j > r_{\max}] = 1 - e^{-x_{r_{\max}}} = 1 - e^{-\frac{\lambda}{2^{r_{\max}}}} \leq 1 - e^{-\frac{\lambda}{2^{\log_2 \frac{\lambda}{\delta}}}} = 1 - e^{-\delta} \leq \delta .$$

□

引理 27。 最大秩 $r_{\max}$ 与 k相距 $O(1)$。

证明。 我们有 $r_{\max} = \lceil \log_2 \frac{\lambda}{\delta} \rceil = \lceil \log_2 \frac{2^k \theta}{\delta} \rceil = k + \lceil \log_2 \frac{\theta}{\delta} \rceil$。因此，$\log_2 \frac{1}{2\delta} < \log_2 \frac{\theta}{\delta} \leq \log_2 \frac{1}{\delta}$，所以 $k + \log_2 \frac{1}{2\delta} < r_{\max} \leq k + \log_2 \frac{1}{\delta}$。□

引理 26 给出了最大寄存器值的一个高概率界。引理 27 表明，满足该界的最大寄存器值与 k相距一个常数。在实践中，对于固定的 m，我们会选择足够小的 δ，使得我们预计任何寄存器中都不会出现高于 $r_{\max}$ 的偏差。δ 的主要作用是控制众数周围带的宽度，以及归入池中的内容。这里重要的是，这一选择本身不依赖于 λ 或 n，而且 $r_{\max}$ 与 k相距一个常数。由于衰减呈双指数级，且左尾有限，因此无需对左尾进行此类截断。

现在，我们将正式定义众数周围的大权重带，以及尾部的小权重池；其边界常数 K_R 和 K_L 将在后文确定。

定义28。 令 $\mathrm{L}_k = \{0, 1, \ldots, k - K_L - 1\}$ 为深左尾部中的索引集合（可能为空），令$\mathrm{R}_k = \{k + K_R + 1, k + K_R + 2, \ldots, r_{\max}\}$ 为深右尾部中的索引集合（可能为空）。池 是索引 $\mathcal{P}_k = \mathrm{L}_k \cup \mathrm{R}_k$的集合。带 由众数 k周围的常宽窗口 $\mathrm{B}_k = \{k - K_L, k - K_L + 1, \ldots, k + K_R - 1, k + K_R\}$ 构成。

定义28指出，带是围绕众数的常宽带，而池则是深尾部中的其余索引。此外，带和池构成索引的一个划分，即 $\mathcal{P}_k \cup \mathrm{B}_k = \mathrm{L}_k \cup \mathrm{R}_k \cup \mathrm{B}_k = \{0, 1, \ldots, r_{\max}\}$ 和 $\mathcal{P}_k \cap \mathrm{B}_k = \mathrm{L}_k \cap \mathrm{R}_k = \emptyset$。需要注意的是，$\mathrm{L}_k$ 或 R_k 可以为空；我们也草率地忽略了索引为零或负数，或超出秩上界的边界情况，但这不会影响结果，因为我们关心的是证明 $|\mathrm{B}_k| = O(1)$，以及池不会对树构造产生实质影响。

我们先估计 L_k 和 R_k的总质量。

引理29. 左尾中的概率和至多为

$$\sum_{\ell \in \mathcal{L}_k} p_\ell \leq 2e^{-2^{K_L}} .$$

证明。 如果$l \in \mathrm{L}_k$，那么根据定义28，$l \leq k - K_L - 1$，因此 $x_\ell = \theta 2^{k-\ell} \geq \theta 2^{K_L+1} > 2^{K_L}$。由于该和的下降速度快于指数级，因此它由首项主导，所以我们可以将整个和界定为$\sum_{l \in \mathrm{L}_k} p_\ell \leq 2e^{-2^{K_L}}$。

□

引理 30。 右尾的概率和至多为

$$\sum_{r\in\mathcal{R}_k} p_r \le 2^{-K_R}.$$

证明。 右尾概率按几何级数递减，且由首项主导，并由引理10将其界定为 2^{-K_R}。□

我们还需要对右尾的最小值以及带进行界定。

引理 31。 对于任意 $k \ge 1$ 和 $\lambda \in I_k$，我们有

- 对于任意 $r \in \mathrm{R}_k$，$p_r \ge g(\delta/2)$，以及
- 对于任意 $b \in \mathrm{B}_k$，$p_b \ge \min\{g(2^{K_L}),\ g(2^{-K_R-1})\}$。

证明。若 $r \in \mathrm{R}_k$，则 $r \le r_{\max}$，且 g 在 $(0,\ \ln 2]$ 上递增，因此 $x_r \ge x_{r_{\max}} > \delta/2$，这意味着对于所有 $r \in \mathrm{R}_k$，$p_r = g(x_r) \ge g(\delta/2)$。

对于第二个断言，若 $b \in \mathrm{B}_k$，则由单峰性可知，从 k向右时概率非增，从 k向左时概率非增，因此最小值在该带的端点处取得。对于左端，我们有 $x_{k-K_L} = \theta 2^{K_L} \le 2^{K_L}$，因此由于 g 在 $[\ln 2, \infty)$ 上递减，我们有 $p_{k-K_L} = g(x_{k-K_L}) \ge g(2^{K_L})$。对于右端，我们有 $x_{k+K_R} = \theta 2^{-K_R} \ge 2^{-K_R-1}$，因此由于 g 在 $(0,\ \ln 2]$ 上递增，我们有 $p_{k+K_R} = g(x_{k+K_R}) \ge g(2^{-K_R-1})$。因此，对所有 $b \in \mathrm{B}_k$，$p_b \ge \min\{g(2^{K_L}),\ g(2^{-K_R-1})\}$。□

现在我们可以确定边界常数 K_R 与 K_L。

引理 32. 存在常数 K_L、$K_R \ge 1$，如定义*28*所述，它们仅取决于 δ，且与 λ 无关，使得

- $\sum_{l\in\mathrm{L}_k} p_l < p_r$ 对每个 $r \in \mathrm{R}_k$都成立，并且
- $\sum_{r\in\mathrm{R}_k} p_r < p_b$ 对每个 $b \in \mathrm{B}_k$都成立。

证明。 对于第一个结论，根据引理29，我们有 $\sum_{l\in\mathrm{L}_k} p_\ell \le 2e^{-2^{K_L}}$；根据引理31，对于每个 $r \in \mathrm{R}_k$，我们有 $p_r \ge g(\delta/2)$。因此，只需选取足够大的 K_L s，使得 $2e^{-2^{K_L}} < g(\delta/2)$。这是可行的，因为 $g(\delta/2)$ 是一个与 K_L 无关的正数，且 $\lim_{K_L\to\infty} 2e^{-2^{K_L}} = 0$。

对于第二个结论，根据引理30，我们有 $\sum_{r\in\mathrm{R}_k} p_r \le 2^{-K_R}$；根据引理31，对于每个 $b \in \mathrm{B}_k$，我们有 $p_b \ge \min\{g(2^{K_L}),\ g(2^{-K_R-1})\}$。因此，只需选取足够大的 K_R s，使得 $2^{-K_R} < \min\{g(2^{K_L}),\ g(2^{-K_R-1})\}$。这是可行的，因为 $g(2^{K_L})$ 和 $g(2^{-K_R-1})$ 是与 K_R 无关的正数，且 $\lim_{K_R\to\infty} g(2^{-K_R-1}) = 0$。□

推论 33. 引理 *32* 表明带和池是良定义的，且带具有与 λ无关的恒定宽度。

推论 34. 引理 *32* 表明，在该分布上应用霍夫曼算法时，左侧尾部首先坍缩为单个节点，随后右侧尾部坍缩为单个节点。左侧尾部与右侧尾部均满足引理 *21* 中的梳状判据，并按预定义合并顺序坍缩为梳状结构。因此，唯一可能发生的非平凡合并是那些涉及带且仅涉及带的合并。

注 35. 我们默认忽略了引理 *32* 中的边界常数可能相当大、且带可能超出秩的下界或上界这一事实，但这并无影响，因为我们只关心渐近地将带界定为恒定宽度。不过，该分布在实践中表现相当良好。

E.4 区间上的树变化

推论 34 表明，可能发生的唯一非平凡合并是涉及常宽带的合并，且只涉及该带，因为尾部在与该带交互之前就会塌缩成梳状树。然而，这还不足以确定区间 I_k上的树变化次数。为此，我们证明所有合并都涉及叶节点的一个子集，因此其权重是指数函数的有限和。注意到，只有当两个权重之差改变符号时，合并顺序才会改变，据此我们可以界定

树变化的次数：只需界定这类和的零点个数。根据罗尔定理，该个数是常数，且与 λ 无关，由此即可证明这一结论。

我们需要以下两个引理，后一个给出了项数有界的指数和的零点个数上界，这是罗尔定理的推论。

引理 36. 对于整数 $1 \le a \le b$,

$$\sum_{r=a}^{b} p_r(\lambda) = e^{-\frac{\lambda}{2^b}} - e^{-\frac{\lambda}{2^a}}.$$

证明。 这直接由第 4.1 节中讨论的分布性质得出。 □

推论 37. 表示折叠的左右尾的节点，其权重是至多由
四个指数函数之和组成。

证明。 观察可知，左尾和右尾都由连续的索引构成；再由引理 36 即可得出该结论。□

引理 38。设 $0 < a_1 < a_2 < \cdots < a_\ell$，且

$$F(\lambda) = \sum_{i=1}^{\ell} c_i e^{-a_i \lambda},$$

使得并非所有 c_i 都为零。那么， F 在$(0, \infty)$上至多有 l 个相异零点。

Proof. 我们对l作归纳。基础情形$l = 1$ 显然成立，因为根据假设， F 没有零点，即 $c_1 \neq 0$.

假设该结论对$l - 1$成立，考虑 $G(\lambda) = e^{a_1 \lambda} F(\lambda)$。显然， G 和 F 有相同的零点，因此只需证明 G 至多有$l - 1$ 个相异零点。根据罗尔定理， G 的任意两个相异零点之间都存在一个 G'的零点。我们有 $G'(\lambda) = \sum_{i=2}^{\ell} c_i (a_1 - a_i) e^{-(a_i - a_1)\lambda}$，它也是由 l 个指数函数构成的和，根据归纳假设，其相异指数至多有 l 个。因此，该结论对 G成立。□

备注 39. 引理 *38* 的结果也可以通过论证 F 在一个紧区间上是解析函数得到。

现在我们可以论证，随着 λ 在区间 I_k上变化，树变化的次数只有常数次。

命题 40. 对于固定的 $k \ge 1$，随着 λ 在 I_k上变化，霍夫曼树至多改变 $O(1)$ 次。

证明. 根据推论 34，尾部具有固定的合并顺序，因此树的结构发生变化的唯一方式，是涉及带的两个候选合并之间出现平局，从而可能改变结构；也就是说，随着 λ 变化，两个候选合并的权重之差改变符号。

由于 $|\mathrm{B}_k| = O(1)$，只有 $O(1)$ 个候选合并可能涉及该带，因此只有$O(1)$ 个候选合并可能改变结构。此外，由于节点的权重是有限项指数和——它们是 B_k的某个子集对应的单元素权重之和，而这样的子集只有常数个，再加上推论 37 中的尾部节点——任意两个候选合并的权重之差都是有限项指数和。因此，根据引理 38，只有 $O(1)$ 个 λ 值可能导致树变化。□

E.5 树变化总数

现在我们来界定树变化的总数，也就是当 λ 上升到某个基数 n的上界 N 时发生的变化。让我们回顾命题 7 的陈述并加以证明。

命题 7. 令 $\lambda_n = \frac{n}{m}$ 对应某个固定的 m，并令 N 为 n的一个上界。当 λ遍历 $\lambda_1, \lambda_2, \ldots, \lambda_N$ 时，树变化的总数为 $O(\log N)$。

证明. 如果 $\lambda \le 1$，那么 $[0, \lambda] \in I_0$，这意味着 $r^*(\lambda) = 1$，由此可知不存在左尾。因此，右尾会衰减成一棵梳状树，至多产生 $O(1)$次树变化。于是，如果 $\lambda_N \le 1$，树变化总数为 $O(1)$，因此假设 $\lambda_N > 1$。

区间 $[0, \lambda_N]$ 至多与 $O(\log N)$个形如 I_k的区间相交，并且根据命题 40， re
最多为 每个这样的区间上有 t $O(1)$ 次树变化，因此树变化的总数为 $O(\log N)$。 □

F 定理 8 和 9 的证明

我们现在分析其余的更新类型。在本节中，我们假设 N 是更新的总数，亦即基数的一个上界，并且 $n = 1, 2, \ldots, N$ 是某一时间步的当前基数。我们固定 m，并设定 $\lambda = \frac{n}{m}$。经过 n 次更新后，我们收到第 $(n+1)^{\text{th}}$ 个元素，其新秩为 $R \sim$ 几何分布$\left(\frac{1}{2}\right)$，且该秩独立于 $R_{\min}$ 或任何当前寄存器值。我们照常将 $x_r = \frac{\lambda}{2^r}$、 $p_0 = e^{-x_0}$ 和 $p_r = e^{-x_r}(1 - e^{-x_r})$ 记为 $r \geq 1$。

令 $R_{\min} = \min_{j \in [B]} R_j$ 为桶中所含的最小秩。以下引理对于分析这些更新是必要的。

引理 41. 对于任意整数 $r \geq 0$，我们有

$$\Pr[R_{\min} \leq r] = 1 - \Pr[R_j > r]^B = 1 - (1 - e^{-x_r})^B ,$$

and

$$\Pr[R_{\min} = r] = \Pr[R_{\min} \leq r] - \Pr[R_{\min} \leq r - 1] = (1 - e^{-x_{r-1}})^B - (1 - e^{-x_r})^B .$$

证明. 根据定义：最小秩至多为 r，当且仅当并非所有寄存器都大于 r。□

当且仅当 $R \geq R_{\min}$ 时才会触发普通更新，否则我们可以直接丢弃该元素。根据 R 与 $R_{\min}$ 的独立性、引理 41 以及对 r 进行边缘化，此类更新的概率为，

$$\begin{aligned}
\Pr[R \geq R_{\min}] &= \sum_{r=0}^{\infty} \sum_{r'=r}^{\infty} \Pr[R_{\min} = r] \Pr[R = r'] \\
&= \sum_{r=0}^{\infty} \Pr[R_{\min} = r] \sum_{r'=r}^{\infty} \Pr[R = r'] \\
&= \Pr[R_{\min} = 0] + \sum_{r=1}^{\infty} \Pr[R_{\min} = r] \Pr[R \geq r] \qquad (9) \\
&= \Pr[R_{\min} = 0] + \sum_{r=1}^{\infty} \Pr[R_{\min} = r] 2^{-r-1} \\
&= \left(1 - (1 - e^{-\lambda})^B\right) + \sum_{r=1}^{\infty} \left((1 - e^{-x_{r-1}})^B - (1 - e^{-x_r})^B\right) 2^{-r-1} .
\end{aligned}$$

现在我们给出普通更新次数的上界。

命题 42. 假设 $m = O(N/\log^3 N)$，经过 N 次插入后，普通更新的期望数量至多为 $O(N/\log N)$。

证明. 为简单起见，令 $\lambda_n = \frac{n}{m}$ 表示 $n = 1, 2, \ldots, N$, $f(x) = (1 - e^{-x})^B$，并将公式 (9) 改写为

$$S(\lambda) = f(\lambda) + \sum_{r=1}^{\infty} \left(f(x_{r-1}) - f(x_r)\right) 2^{-r-1} . \qquad (10)$$

我们需要计算这个求和

$$\sum_{n=0}^{N} S(\lambda_n) . \qquad (11)$$

为此，我们将对公式 (10) 的内层求和进行界定。将公式 (10) 分为三部分：常数项 $f(\lambda)$、前几个秩 $1 \leq r < r^* = \lceil \log_2 \lambda \rceil$，以及其余各项 $r \geq r^*$:

$$S(\lambda) = f(\lambda) + \sum_{r=1}^{r^*-1} \left(f(x_{r-1}) - f(x_r)\right) 2^{-r-1} + \sum_{r=r^*}^{\infty} \left(f(x_{r-1}) - f(x_r)\right) 2^{-r-1} . \qquad (12)$$

根据伯努利不等式，公式（12）的第一项有如下界：

$$1 - (1 - e^{-\lambda})^B \leq Be^{-\lambda} \leq \frac{B}{\lambda} \qquad (13)$$

对于第二项，注意到 f 递增，且 $f(x_{r-1}) \le 1$，因此

$$f(x_{r-1}) - f(x_r) \le 1 - f(x_r) = 1 - (1 - e^{-x_r})^B \le Be^{-x_r},$$

因此

$$\sum_{r=1}^{r^*-1} \left(f(x_{r-1}) - f(x_r)\right) 2^{-r-1} \le \frac{B}{2} \sum_{r=1}^{r^*-1} e^{-x_r} 2^{-r} = \frac{B}{2} \sum_{r=1}^{r^*-1} e^{-\frac{\lambda}{2^r}} 2^{-r}. \tag{14}$$

代入 $k = r^* - r$，可得 $2^{-r} e^{-\frac{\lambda}{2^r}} = 2^{-r^*} 2^k e^{-\frac{\lambda}{2^r}}$。此外，我们观察到 $\frac{1}{2} < \frac{\lambda}{2^{r^*}} \le 1$，这意味着 $e^{-\frac{\lambda}{2^r}} \le e^{-2^{k-1}}$，因此可以进一步对公式（14）的和进行如下界定：

$$\sum_{r=1}^{r^*-1} \left(f(x_{r-1}) - f(x_r)\right) 2^{-r-1} \le \frac{B}{2} 2^{-r^*} \sum_{k=1}^{\infty} e^{-2^{k-1}} 2^k = O\left(\frac{B}{\lambda}\right), \tag{15}$$

这是因为该级数收敛于一个常数，且 $r^* = \lceil \log_2 \lambda \rceil$。

最后，对于公式（12）的第三项，当 $r \ge r^*$ 时，我们有 $x_r \le 1$，因此 $1 - e^{x_r} \le x_r$，进而 $f(x_r) = (1 - e^{-x_r})^B \le x_r^B$，当 $r \ge r^*$ 时，因此 $f(x_{r-1}) - f(x_r) \le f(x_{r-1}) \le x_{r-1}^B$。因此，

$$\sum_{r=r^*+1}^{\infty} \left(f(x_{r-1}) - f(x_r)\right) 2^{-r-1} \le \sum_{r=r^*+1}^{\infty} \left(\frac{\lambda}{2^{r-1}}\right)^B 2^{-r-1} = O\left(\frac{1}{\lambda}\right),$$

因为 $2^{r^*-1} < \lambda \le 2^{r^*}$，所以该级数为 $O(2^{-r^*}) = O\left(\frac{1}{\lambda}\right)$。

我们需要检查 $r = r^*$ 的特殊情况，此时有 $0 \le \left(f(x_{r^*-1}) - f(x_{r^*})\right) 2^{-r^*-1} \le 2^{-r^*-1} = O\left(\frac{1}{\lambda}\right)$。

综合以上结果，可知公式（12）的内层求和至多为 $S(\lambda) = O\left(\frac{B}{\lambda}\right) = O\left(\frac{Bm}{n}\right)$，因此我们得到

$$\sum_{n=0}^{N} S(\lambda_n) = O\left(Bm \log N\right) = O\left(m \log^2 N\right),$$

通过用自然对数近似调和级数，并考虑到 $B = O(\log N)$。最后，如果 $m = O(N/\log^3 N)$，那么普通更新的期望次数至多为 $O(N/\log N)$，从而得证。□

推论 43. 假设一次普通更新需要 $O(\log n)$ 时间，且 $m = O(n/\log^3 n)$，那么一次普通更新的摊销成本在 n 次更新中为 ***O(1)***。

推论 43 符合直觉：只有当元素数量相对于寄存器数量足够大时，普通更新的期望次数才会被限制在合理范围内；否则几乎所有寄存器都是空的，此时使用线性计数 [WZT90] 更合适。

接下来，我们分析最小秩更新的次数。

命题 44. N 次插入后，最小秩更新的期望次数至多为 $O(m/\log N)$.

证明。共有 $O\binom{m}{B} = O\left(\frac{m}{\log N}\right)$ 个桶，且可能的秩值有 $O(\log N)$ 个。因此，可能的最小秩更新至多有 $O(m)$ 个。□

推论 45. 假设一次最小秩更新需要 $O(\log^2 n)$时间，且 $m = O(n/\log^3 n)$，则最小秩更新的摊销成本在 n 次更新中为 ***O(1)***。

推论 45 的结论相当宽松，但由于普通更新的成本已涵盖这一点，因此足以满足我们的需求。现在我们可以将这些结果综合起来，得到以下结论。让我们回顾定理 8 的陈述并加以证明。

定理 8. 假设一次普通更新需要 $O(\log n)$ 时间，一次最小秩更新需要 $O(\log^2 n)$ 时间，而一次霍夫曼树更新后再对所有码字重新编码需要 $O(m \log n)$ 时间；此外，假设 $m = O(n/\log^3 n)$，则在 n 次更新中，更新的摊销成本为 ***O(1)***。

证明。 由于我们要界定 n 次更新的摊销成本，只需分别界定所有这些成本，再将各个上界相加。根据命题 7，树更新次数至多为 $O(\log n)$，因此树更新的总成本至多为 $O(m \log^2 n)$；根据关于 m 的假设，这一上界为 $O(n/\log n)$。根据命题 42，普通更新的总成本至多为 $O(n)$；根据命题 44，最小秩更新的总成本至多为 $O(m \log n)$，根据关于 m 的假设，这一上界为 $O(n/\log^2 n)$。因此，所有更新的总成本至多为 $O(n/\log n + n + n/\log^2 n) = O(n)$，故摊销成本为 O(1)，证毕。□

让我们回顾定理 9 的陈述并予以证明。

定理 9. 假设我们可以在常数时间内访问 $\Omega(\log n)$ 位以及 $m = \Omega(\log^2 n)$，从而可以维护霍夫曼码本，因此可以假设一次普通更新耗时 $O(1)$，一次最小秩更新耗时 $O(\log n)$，一次霍夫曼树更新及随后对所有码字的重新编码耗时 $O(m)$，则在 n 次更新上的摊销成本为 *O(1)*，如果我们进一步假设 $m = O(n/\log^2 n)$。

证明. 根据命题 7，树更新的次数至多为 $O(\log n)$，因此树更新的总成本至多为 $O(m \log n)$，即 $O(n/\log n)$（在关于 m 的假设下）。普通更新的总成本至多为 $O(n)$（根据命题 42，若针对 m 进行调整），最小秩更新的总成本至多为 $O(m)$（根据命题 44），即 $O(n/\log^2 n)$（在关于 m 的假设下）。因此，所有更新的总成本至多为 $O(n)$，故摊销成本为 O(1)，如所述。□

G HBS 草图操作的伪代码

HBS 草图操作的伪代码见算法 1。

算法 1 哈夫曼桶草图操作。

函数 PEEK(b, j)
    使用一元数组确定桶 b 中的 j^{th} 码字起始位置 s_j 和长度 l_j
    从码字数组中的位 $s_j, s_j + 1, \ldots, s_j + l_j - 1$ 获取码字 c_j
    使用霍夫曼树或查找表解码 c_j，得到秩值 r
    返回 r
结束函数
函数 POKE(r, b, j)
    使用霍夫曼树或查找表将 r 编码为码字 c
    将 c_{old} 替换为 c，必要时移动桶
结束函数
函数插入(y)
    计算 $b \leftarrow h_b(y)$、$j \leftarrow h_j(y)$ 和 $r \leftarrow \rho(h_r(y))$。
    如果 $r > r_{\min}$ 属于桶 b **则**
        $r_{\text{old}} \leftarrow \text{Peek}(b, j)$
        如果 $r > r_{\text{old}}$ **则**
            Poke(r, b, j)
            如果 $r_{\text{old}} = r_{\min}$ **则**
                $c_{\min} \leftarrow c_{\min} - 1$
                如果 $c_{\min} = 0$**那么**
                    通过 Peek 桶中的所有寄存器，重新计算 $r_{\min}$ 和 $c_{\min}$
                结束条件判断
            结束条件判断
            更新 $\hat{n}_b$和 $\hat{n}_{\circ}$
            如果 $\hat{n} - \hat{n}_{\text{old}}$ 过大**则**
                重构霍夫曼树并重新编码所有桶
                $\hat{n}_{\text{old}} \leftarrow \hat{n}$
            结束 if
        结束 if
    结束 if
结束函数
函数 合并(S_1, S_2)
    对于所有 $b \in [m/B]$ **执行**
        通过遍历 $j \in [B]$ 并取草图中 r_j 的最大值来构造 $\hat{n}_b$
    结束循环
    $\hat{n} \leftarrow \sum \hat{n}_b$
    如果 $\hat{n} - \max\{\hat{n}_{\text{old},1}, \hat{n}_{\text{old},2}\}$ 太大 **则**
        使用 $\hat{n}$ 构造霍夫曼树
    else
        复用 $\hat{n}_{\text{old}}$ 较大的草图中的霍夫曼树
    结束 if
    对所有 $(b, j) \in [m/B] \times [B]$，**执行**
        $r \leftarrow \max\{\text{Peek}^{S_1}(b, j), \text{Peek}^{S_2}(b, j)\}$
        Poke(r, b, j)
    结束 for
结束函数