arXiv:2303.04722v1 [cs.DS] 8 Mar 2023

修订版 B-Treap：编写高负载下高效的随机块搜索树

RoodabehSafavi ！ 奥地利科学技术研究所（ISTA），Am Campus 1，A-3400 克洛斯特新堡，奥地利

MartinP.Seybold ！ 维也纳大学计算机科学学院，算法理论与应用系，韦林格大街 29 号，A-1090 维也纳，奥地利

摘要

唯一表示数据结构为每个逻辑状态对应唯一的存储状态。我们研究在此背景下，如何维护一个包含来自全序宇宙的 n 个键的动态集合。

我们提出一种称为（α, ε）随机块搜索树（RBST）的双层数据结构，它具有唯一表示且适用于外部存储器。尽管 RBST 自然地推广了广为人知的二叉Treap，但要分析其以块读取、块写入和存储块数衡量的期望搜索、更新和存储效率，还需要一些新思路。我们证明，搜索需要 $O(\varepsilon^{-1} + \log_\alpha n)$ 次块读取；对于每个 $\varepsilon \in (0, 1/2]$，(α, ε)-RBST 的渐近负载因子至少为 $(1-\varepsilon)$；动态更新需要 $O(\varepsilon^{-1} + \log_\alpha(n)/\alpha)$ 次块写入，即当 $\alpha = \Omega(\log n \log\log n)$ 时，只需 $O(1/\varepsilon)$ 次写入。因此，相较于已知的唯一表示 B-Treap [Golovin；ICALP'09]，(α, ε)-RBST 在搜索效率、存储效率和写入效率方面提供了更优的界。

2012 ACM主题分类 计算理论 → 算法设计与分析

关键词与短语 唯一表示、随机化、分块搜索树、写入效率、存储效率、计算几何、自顶向下分析

资助 本研究获得澳大利亚研究理事会探索项目资助计划的支持（项目编号 DP180102870）。此外，本研究还获得欧洲研究理事会通过欧盟地平线2020研究与创新资助计划提供的资助（资助编号 101019564“现代全动态数据结构的设计”）。

1 引言

以支持插入、删除和搜索的方式组织一组键，是计算机科学中最基础的问题之一，它推动了具有各种保证和性能权衡的搜索结构的发展与分析。例如，二叉搜索树已知有多种高度平衡方法（如 AVL 树和红黑树）以及重量平衡方法（如 BB[a]树），其中一些每次更新只需 $\mathcal{O}(1)$ 次写入（例如 [9]）。

块搜索树是二叉搜索树的推广形式，每个树节点中最多存储 α 个键和 $\alpha+1$ 个子指针，其中块的数组大小 α 是一个（通常为）固定参数。由于这类布局能够保证一定的数据局部性，因此块结构对于具有存储层次结构的机器模型中的高效读写访问至关重要。

外部存储器与确定性 B 树的变体 在经典的外部存储器（EM）模型中，问题实例的 n 数据项必须以固定大小 B的块，在块可寻址外部存储器与能够存储多达 M 个数据项并执行计算的内部主存（MM）之间传输。通常，$n \gg M > B$ 以及主存在整个处理过程中只能存储少量块，例如 $M/B = \mathcal{O}(1)$。尽管 Vitter 的并行磁盘模型可以形式化更一般的设置（见 [21, 区段 2.1]），但我们关注最基本的情形：一个仅支持块寻址的外部存储设备，以及一个

带主存的计算设备。计算成本通常按外部存储器的块读取次数（Is）、块写入次数（Os），或在外部存储器上执行的 IO 总次数来衡量。对于范围搜索等基本算法任务，使用块搜索树并令 $\alpha = \Theta(B)$ 即可获得总 IO 次数渐近最优的算法（例如，见综述中的区段 10 ） [21]。

经典 B 树 [3] 保证除根节点外，每个块的负载因子至少为 $1/2$ ，且所有叶节点的深度相同，这依赖于确定性的平衡准则。B*树变体 [14, 第 6.2 节。4] 通过一种（确定性的）溢出策略保证负载因子至少为 $2/3$ ，该策略会在同一层的两个相邻节点之间共享键。Yao 的边缘分析 [22]表明，在随机顺序下插入键到（确定性平衡的）B 树中，预期渐近负载因子为 $\ln(2) \approx 69\%$ ；而对于 B* 树，预期渐近负载因子为 $2\ln(3/2) \approx 81\%$。然而，对于一般工作负载，维持更高的负载因子（采用更宽泛的溢出策略）会进一步增加分块搜索树中更新的写入开销 [2, 15]。

广受欢迎的写入优化型 B$^\varepsilon$树 [7, 4]（例如 [10, 13], ）在每次插入的摊销块写入次数 $\mathcal{O}(\log_{1+\alpha^\varepsilon}(n)/\alpha^{1-\varepsilon})$ 与搜索的块读取次数 $\mathcal{O}(\log_{1+\alpha^\varepsilon} n)$ 之间提供了平滑的权衡。例如，调优参数 $\varepsilon = 1$ 可保留 B 树的界，而 $\varepsilon = 1/2$则能提供更优的界。然而，在完全写入优化的情形下（$\varepsilon = 0$），搜索的块读取次数上界仅为 $\mathcal{O}(\log n)$。实现这一点的主要设计思路，是为 B 树的非叶节点增设“消息缓冲区”，使键的插入操作能够存储在靠近根节点的位置。只有当消息缓冲区满时，消息才需要继续向树的下层传递（参见 [7, 区段 3.4] 和 [4]）。

显然，这两种（确定性）结构的状态很大程度上取决于键实际插入其中的顺序。

唯一表示数据结构 出于安全或隐私方面的原因，某些应用（例如参见 [11], ）要求数据结构不能向内存表示的观察者泄露任何有关历史状态的信息，即不能泄露已删除键的证据，也不能泄露键的插入顺序证据。如果数据结构具有这种强历史独立性，就称其为唯一表示（UR）数据结构。例如，有序数组是 UR 数据结构，但上述搜索树并非 UR 数据结构，这是因为其中存在确定性的再平衡等操作。早期研究 [19, 20, 1] 给出了指针机上 UR 数据结构中基于比较的搜索的下界和上界。利用 RAM 模型中的 UR 哈希表 [5, 17] ，还可以将指针结构唯一地映射到内存中。（参见 [5] 中的定理 4.1 以及 [6]中的第 2 节。）

随机化搜索树的定义是：按照随机排列的顺序，逐个将键插入一棵初始为空的二叉搜索树中。著名的 Treap [18]维护由排列顺序决定的树形结构，并支持高效的插入、删除和搜索（另见 [16, 第 1 章3]）。每次插入或删除执行期望为$\mathcal{O}(1)$ 次旋转（写操作），而搜索读取的树节点数期望最多为 $2\ln(n+1)$，这对于具有非对称成本或并发访问的机器模型尤为值得注意。

Golovin [12]提出了 B-Treap，将其作为首个用于字节可寻址外部存储器（BEM）的 UR 数据结构。B-Treap 源于对关联（二叉）Treap 采用特定的块存储布局。也就是说，块树通过自底向上的方式迭代构建：将特定大小的子树放入一个块节点中[1]。作者在特定条件下给出了 B-Treap 的界

[1] 例如，块内子节点指针需要使用字节可寻址内存，并且每个键都维护一个大小字段，用于存储其子树大小（在关联的二叉 Treap 中）。

	数据结构	模型	# 块	
非 UR	B 树 [3]	EM	Size	$\mathcal{O}(n/\alpha)$
			更新	$\mathcal{O}(\log_\alpha n)$
			搜索	
	B^ε-树 [7]	EM	Size	$\mathcal{O}(n/\alpha)$
			更新	$\mathcal{O}\left(\frac{1}{\varepsilon}\frac{\log_\alpha n}{\alpha^{1-\varepsilon}}\right)$
			搜索	$\mathcal{O}\left(\frac{1}{\varepsilon}\log_\alpha n\right)$
UR	B-Treap [12]	BEM	Size	$\mathcal{O}(n/\alpha)$
			更新写入	$\mathcal{O}\left(\frac{1}{\varepsilon}\log_\alpha n\right)$
			搜索读取	
	(α,ε)-随机二叉搜索树	EM	Size	$\mathcal{O}(n/\alpha)$ $\leq (1+\varepsilon)n/\alpha$
			更新写入	$\mathcal{O}\left(\frac{1}{\varepsilon}+\frac{\log_\alpha n}{\alpha}\right)$ $\mathcal{O}(1/\varepsilon)$
			搜索读取	$\mathcal{O}(\frac{1}{\varepsilon}+\log_\alpha n)$

表 1 外存中已知的非 UR 和 UR 数据结构以及本文提出的随机二叉搜索树概览。

参数条件：若 $\alpha=\Omega\left((\ln n)^{\frac{1}{1-\varepsilon}}\right)$，其中 $\varepsilon>0$ 为某个值，则更新的期望块写入次数为 $O(\frac{1}{\varepsilon}\log_\alpha n)$，范围查询的期望块读取次数为 $O(\frac{1}{\varepsilon}\log_\alpha n+k/\alpha)$，其中 k 为输出大小。若 $\alpha=\Omega\left(n^{\frac{1}{2}-\varepsilon}\right)$，其中 $\varepsilon>0$ 为某个值，则深度以高概率为 $O(\frac{1}{\varepsilon}\log_\alpha n)$，且存储空间为线性（见 [12]中的定理 1）。论文还讨论了来自非动态实现的实验数据，表明期望负载因子接近 $1/3$（见 [12]中的第 6 节）。尽管存储布局方法能够利用已知的 Treap 界来分析 B-Treap，但块大小条件（例如 $\alpha=\Omega(\ln^2 n)$）限制了 B-Treap 在算法中的应用。

1.1 贡献与论文结构

我们提出了一种名为$(\alpha,\ \varepsilon)$-随机化块搜索树（RBST）的双层随机数据结构，其中 α 是块大小，而 $\varepsilon\in(0,\ 1/2]$ 是用于提高次级层空间效率的调节参数（见第2节）。去掉辅助结构后，RBST 的树分布恰好等同于按随机顺序逐个插入键、构建初始为空的分块搜索树所得到的树分布（例如，$\alpha=1$ 会得到 Treap 分布）。RBST 是 UR 的，搜索算法也很简单。我们给出了一种用于动态更新的部分重建算法，该算法占用$\mathcal{O}(1)$ 块的临时内存存储，并且其对外存执行的写操作次数与 RBST 的结构变化成正比（见第2.1节）。

我们针对 (α,ε)-随机二叉搜索树，证明了关于块读取、块写入和所存储块数的三项性能界。第 3 节表明，搜索期望读取 $O(\varepsilon^{-1}+\log_\alpha n)$ 个块。第 4 节表明，对于每个 $\varepsilon\in(0,1/2]$，(α,ε)-随机二叉搜索树的期望渐近负载因子至少为 $(1-\varepsilon)$。结合这两种分析方法，我们在第 5 节表明，动态更新执行时的块写入次数期望为 $O(\varepsilon^{-1}+\log_\alpha(n)/\alpha)$。

因此，对于所有块大小 $\alpha\geq 1$，随机二叉搜索树都是简单的 UR 搜索树；与自底向上的

B-Treap [12]。参见表 1，其中比较了适用于外存的已知 UR 和非 UR 数据结构。据我们所知，(α, ε)-随机二叉搜索树是首个同时提供最优搜索界、完全具备写入效率和存储效率的数据结构。我们的分块搜索树设计的核心是次级层，它为块节点采用与子树权重成正比的扇出，也就是说，与其中存储的键的数量成正比。

2 随机化块搜索树

与众所周知的 B 树不同，后者在每个块中存储 α 到 $\alpha/2$ 个键，并且所有叶节点的深度都相同；我们的 (α, ε)-随机二叉搜索树定义并不严格要求每个节点都达到最低负载。对于键集合 X（包含 n 个键），随机二叉搜索树 $T_X(\pi)$的形状由一个增量过程定义：按照随机顺序 π，将键 $x \in X$ 插入初始为空的分块搜索树结构中，即优先级值按升序排列 $\pi(x)$。动态更新的具体算法将在第 2.1 节讨论。接下来，我们介绍同时支持报告查询（即范围搜索）和（顺序）统计查询（即范围计数，以及搜索第 k 小的值）的基本树结构。在第 2.1 节中，我们还将讨论如何省去显式存储子树权重；当应用不需要统计查询时，这样做可以改进更新界。接下来，我们定义分块搜索树的结构并陈述其不变式。

设数组大小 $\alpha \geq 1$ 和缓冲区阈值 $\beta \geq 0$ 为固定的整数参数。每个随机二叉搜索树块节点都存储一个固定大小的数组，用于存放 α 个键和 $\alpha + 1$ 个子节点指针，以及一个父节点指针和该节点到根节点的距离。我们使用节点数组中所含键里具有最小优先级 $\pi(x)$ 的键值 x 作为块节点 UR 的标签。每个子节点指针均由一对值组成，分别存储指向子块的指针，以及该子树中包含的键总数 $s \geq 1$。（例如，父节点无需读取相应的子节点，就能立即获知该节点子树的权重。）随机二叉搜索树中的所有内部块均为满块，即恰好存储 α 个键。每个块有两种可能的状态之一：非缓冲（$s \geq \beta$）或 缓冲（$s < \beta$）。虽然 (α, ε)-随机二叉搜索树将使用 $\beta = \Theta(\alpha^2/\varepsilon)$，但我们首先给出不使用缓冲区（$\beta = 0$）的主树定义，即每个块都保持非缓冲状态。

对于 $\beta = 0$，树由以下增量过程定义：按照优先级 $\pi(\cdot)$升序插入键。从根块开始，插入键 x时，首先根据搜索树性质定位树的叶块。如果叶块未满，则将 x放入该节点的有序键数组中。否则，会（根据搜索树性质）分配一个新的子块，并将 x 存储于其中。因此，任何内部块都存储其子树中优先级值最小的那些 α 键[2]。图 1 (左)给出了一个包含 26 个键的示例，其中 $\beta = 0$ 由 12 个块组成，这表明叶节点的块中可能仅存储一个键。我们的树结构通过以下方式解决了这个问题。

对于 $\beta > 0$，核心思路是等到子树中包含足够多的键后，再生成新的叶节点。为此，我们使用称为缓冲区的辅助结构，替代所有包含 $\leq \alpha + \beta$ 个键的子树的存储布局。因此，主结构中其余的块节点都已填满，其子节点要么是一个主块，要么是一个缓冲区。缓冲区的 UR 组织方式有多种，例如，使用一个至多包含 $\lceil(\alpha + \beta)/\alpha\rceil = \mathcal{O}(\alpha/\varepsilon)$ 个按键值排序的块的列表。我们提出以下 UR 结构，以获得更紧的界。

[2] 例如，$\alpha = 1$ 和 $\beta = 0$ 可得到著名的 Treap，而 $\alpha > 1$ 可得到扇出为 $\alpha + 1$的搜索树。

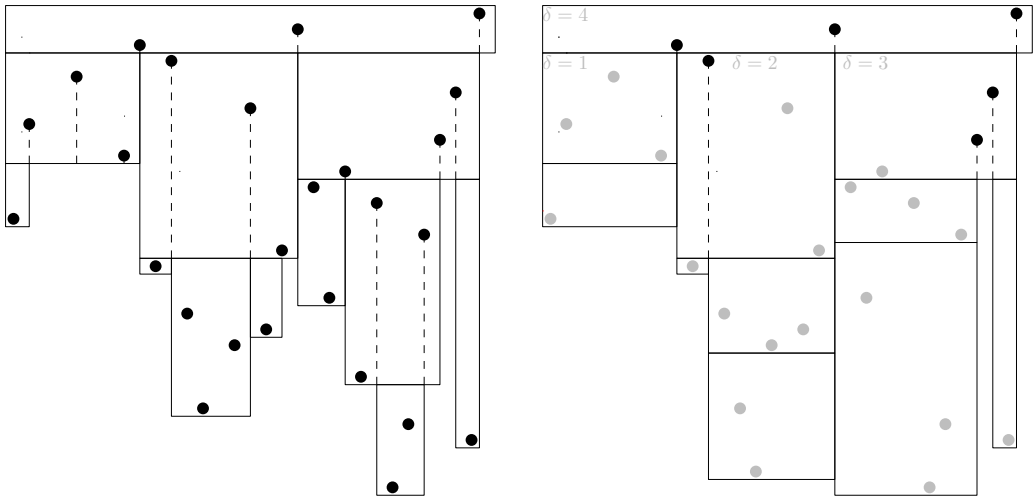

图 1 两个随机二叉搜索树的笛卡尔表示，展示了平面中的点 $(i,\ -\pi(i))$，该点对应第 i 大的键 $x_i \in X$，其优先级值为 $\pi(i)$（$n = 26$ 和 $\alpha = 3$）。每个块 v 的活动分隔符以竖线表示，水平线 $p^\dagger(v)$。左侧的树具有 $\beta = 0$，右侧的树具有缓冲子树。注意，根到叶路径上的所有块（最后一个块除外）都是满的。

次级 UR 树：扇出与权重成比例的缓冲区 我们的缓冲区是搜索树，由类似于主树的节点组成；这些节点也会在其块中存储来自子树的 α 个优先级值最小的键。不过，内部节点的扇出会根据子树中的键数而变化。定义 $n \le \alpha + \beta\ \delta(n) = \min\{\alpha+1, \max\{1, \lceil \frac{n-\alpha}{\rho} \rceil\}\}$ 作为权重为 $n \le \alpha+\beta =$ 的子树的扇出上界：$\alpha + (\alpha+1)\rho$。参数 $\rho = \Theta(\alpha/\varepsilon)$ 的取值将在第 4 节中确定。为构造 UR 结构，缓冲区的布局再次仅根据缓冲区中的键集合以及将这些键映射到优先级值的随机排列 π 递归定义。

对于 $\delta = 1$，子树根节点至多有一个子节点，因此形成一个块列表。每个块中的键按键升序存储，列表中的块则按优先级值升序排列。也就是说，第一个块包含优先级最小的那些键；第二个块从剩余的 $n - \alpha$ 个键中，包含优先级值最小的那些键；依此类推。

对于 $\delta \in [2, \alpha+1]$，我们还会将优先级最小的 α 个键存储在根节点中。我们将优先级值最小的 $\delta - 1$ 个键称为该块的 活动分隔符 。子树中剩余的 $n - \alpha$ 个键仅使用活动分隔键划分为 δ 组键值。每组键都在缓冲区中递归组织，由此形成的树则作为活动分隔键的子节点。

图1 右侧给出了同一组（α，ε）-随机二叉搜索树的示例，这组点由26个点组成，分为 11 个块。关于主级和次级随机二叉搜索树节点的定义，有几点需要说明。由于存储布局仅取决于键的优先级顺序 π 、缓冲区中的键数以及其中键值的顺序，因此该缓冲区是 UR。注意，对子指针的状态值求和即可得到 $s_1 + \ldots + s_\delta + \alpha = n$子树的权重，无需额外的块读取。此外，每当次级结构中的更新使缓冲区的根节点超过阈值，即 $s_1 + \ldots + s_{\alpha+1} > \beta$，其根节点所包含的键便是该集合中优先级 α 最小的那些键，且所有键都是活动分隔键；也就是说，缓冲区根节点会立即满足主树中块的两个不变式。

请注意，这种随机二叉搜索树的定义不仅自然涵盖块大小为 α的随机化搜索树，还允许平滑地权衡，逐步趋向于简单的块列表（分别为 $\varepsilon = \infty$ 和 $\varepsilon = 0$）。尽管叶节点非空，随机二叉搜索树中的叶节点负载仍可能低至 $1/\alpha$ 。尽管上述分块搜索树的定义具有过程性，但它并未给出实际构建和维护随机二叉搜索树所需的空间和 I/O 高效算法。本区段余下部分将说明用于计算随机二叉搜索树的搜索、动态插入和动态删除算法。我们的分析见第3、4和5节。

在 (α, ε)-随机二叉搜索树中的后继与范围搜索

与普通 B 树一样，我们通过对数组进行二分搜索来确定搜索值 q 的后继，即块中值不小于q的最小活动分隔键，并沿相应的子指针前进。如果当前搜索节点正在缓冲（即活动分隔键少于 α 个），我们还会在搜索过程中检查块中非活动键的集合，并将其存放在局部主存变量中，以找到 q 的后继。搜索会向下进行，直至找到相应随机二叉搜索树缓冲区中扇出为 ≤ 1 的列表（即块节点）。由于列表按优先级排序，我们通过扫描列表中的 $\mathcal{O}(1/\varepsilon)$ 个块来检查所有键，从而确定 q 的后继。按照通常方式，将范围的两个边界 $[q, q']$ 的搜索路径纳入考量，即可将此点搜索扩展为查询范围搜索。

总之，后继与范围搜索在主存中占用$\mathcal{O}(1)$个块，不写入块，并且块读取次数为$\mathcal{O}(D)$，其中 D 是随机二叉搜索树的深度。

2.1 通过部分重建进行插入和删除

与 Treap 树一样，(α, ε)-随机二叉搜索树的树形结构完全由键的排列π 决定。对于 $X \cup \{x\} = X'$，任何将 $T_X(\pi)$ 转换为 $T_{X'}(\pi)$ 或反向转换的更新方法，都是适用于插入和删除的算法。不同于总是在叶层执行插入和删除、并使用旋转的 Treap 树，在分块搜索树中实现和分析旋转似乎颇具挑战。我们提出了一种以自顶向下方式重建子树的更新算法，该算法与替罪羊（二叉搜索）树的重新平衡机制存在某些模糊的相似之处。

一种朴素的更新算法可能会尝试完全重建其数组因键插入或删除而发生变化的块所对应的整个子树。这也有助于维持节点与根节点之间的距离（以证明搜索性能）。更新算法面临的主要困难，是为外存访问实现与树中的结构变化（近似）成比例的 I/O 上界，同时在$\mathcal{O}(1)$ 的最坏情况上界下，控制主存中所需的临时块数量。本区段介绍的部分-重建算法，根据更新的结构变化，对块读取（Is）和块写入（Os）的数量给出了界；随机二叉搜索树更新的预期结构变化将在第5节中进行分析。

▶ 观察*1.* 设 π 是键全集 U 上的一个排列，$X' = X \cup \{x\} \subseteq U$ 分别表示插入 x 后和插入前树中的键。设 m 为随机二叉搜索树 $T_X(\pi)$ 中与随机二叉搜索树 $T_{X'}(\pi)$ 中不同的块数，m' 为 $T_{X'}(\pi)$ 中与 $T_X(\pi)$ 中不同的块数。设 D 和 D' 分别为 $T_X(\pi)$ 和 $T_{X'}(\pi)$ 中包含这些块的子树的高度。本区段中的更新算法在更新期间向外存执行 $O(m + m')$ 次块写入，读取 $O(m' + D' \cdot m)$ 个块，并使用 $O(1)$ 个块的临时内存存储。

我们的重建算法旨在尽可能减少一次更新中的块写入次数，采用贪心自顶向下方法，在辅助外存储中构建 $T_{X'}$ 的子树，同时

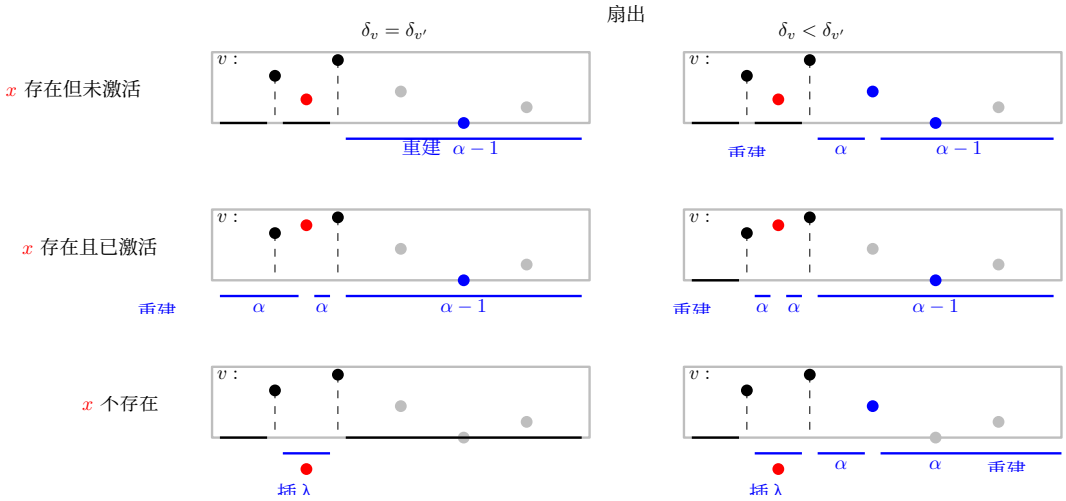

图 2 在随机二叉搜索树的块 v 中插入键 x （红色）的重建情形。

从 UR 外存读取 T_X 中的键。完成后，我们从 UR 内存中删除 m 废弃块，并将 m' 结果块写回 UR，以获得最终树 $T_{X'}$。这样，我们的更新算法只需在内存中使用 O(1) 个块的临时存储（并且在子树重建期间仍支持搜索）。基本思路如下。在组装块 v' 并写入辅助外存时，给定指定的分隔符区间 $(\ell(v')\ r(v'))$，我们通过在 T_X 中搜索，确定优先级值最小的 α 个键，并将它们放入 v' 的数组中；随后确定 v' 块节点的扇出 $\delta_{v'}$ 和活动分隔键，并递归构建 v' 中活动分隔键之间每个区段内剩余键对应的子树。最后，我们释放 T_X 在 UR 中的废弃块，并将结果子树从辅助外存移至 UR 内存。接下来，我们讨论向树中插入新键 x 时可能出现的重建情形，具体实现将在后文讨论。对于块节点 v，令 $X(v)$ 表示存储在 v 子树中的键集合，$B(v)$ 表示存储在 v 本身数组中的键，$\subseteq X(v)$ 为 $\min\{\pi(y) : y \in B(v)\}$，而 $p^*(v)$ 为 $\max\{\pi(y) : y \in B(v)\}$。（例如，键 y 的 $\pi(y) = p^*(v)$ 是块 v 的 UR 标签。）

考虑在随机二叉搜索树 T_X中搜索 x 的路径。从根节点开始，在遇到第一个节点 v 时停止，该节点 i) 必须将 x 存入其数组（即 $\pi(x) \in [p^*(v)\ p^\dagger(v)]$），或 ii) 需要重建（即必须增加其扇出）。为检查情形 ii)，我们使用 v 的增强子节点引用，确定结果子树 v'中的键数。因此，无需额外读取块，即可根据存储的子树权重得知插入前的扇出 δ_v 和 $\delta_{v'}$ 插入后的扇出[3]。

如果 i) 和 ii) 均未发生，搜索会在缓冲区列表（$\delta_v = \delta_{v'} \leq 1$）中终止，我们只需插入该键，并通过一次线性遍历重写列表中所有后续块。请注意，如果省略该列表（因为活动分隔键区间内没有键），则只需分配一个新块来单独存储 x，以生成输出随机二叉搜索树。接下来说明情形 i) 和 ii) 的插入算法。所有情形的概览见图 2 。

在 ii) 中，v 是一个需要增加扇出的缓冲节点。只需从一棵子树构建两棵树，使它们各自的根节点包含顶部的 α 个键。如果新键 x 包含在

[3] 为避免显式存储和更新子树权重，可以沿着 x 的搜索路径自底向上计算子树权重，直到找到包含 m' 块的节点 v。

此区间内，我们就将 x 作为额外的“传递”参数传给过程。如果 x 不包含在此区间内，则沿着 v 中必定包含 x的相应子节点继续插入。（参见图2右下角。）注意，在辅助外存中最多会构建 v' 的三棵子树，并且读取操作最多发生在 v的两棵子树中，无论 α的实际值如何。

在 i) 中， v 是一个存储 α 个键的内部节点，共有四种情况，具体取决于 x是否是 v' 中的活动分隔键，以及扇出是保持不变还是增加（见图2）。在所有情况下，最多只需重建三棵子树：一棵是与 x 相邻的两个分隔区间对应的树；另一棵包含与被插入 y 而从块 v 中移出的键 x相邻的两个分隔区间所对应的子树，即 $\pi(y) = p^{\dagger}(v)$。请注意，判断向作为缓冲块的块 x 中插入 v 是否会增加扇出，无需额外的块读取，因为相应子树的权重与子节点引用一同存储在 v中。

请注意，从随机二叉搜索树中删除键 x 时，重建活动分隔键之间键区段的情况与插入时相同。此外，如果删除导致产生缓冲块（$\delta_{v'} < \alpha+1$），则其所有子节点也都是缓冲块，因此会将各自的子树权重与其引用一同存储。因此，确定删除操作的扇出同样无需额外的块读取。

为了完成更新算法，我们还需要说明如何针对两个活动分隔键之间的给定区间执行自顶向下贪心重建过程。首先，我们将查找相关键的任务单独处理；这些键用于在以下范围搜索函数 $\texttt{top}(k, v, (\ell, r))$ 中，将下一个块写入辅助外存。给定子树根节点 v（位于 T_X 中）、正整数 $k \leq \alpha$ 以及查询范围 (ℓ, r)，我们报告 $Y = X(v)$ 中那些属于 $\cap(\ell, r)$ 的键：它们具有 k 个最小的优先级值，并附上各自的子树权重 $w_0 + \ldots + w_k = |Y|$。为此，采用朴素搜索，按键升序访问与查询范围重叠的 v 的所有后代节点。

对于 α 重建或 $\alpha - 1$ 重建，我们分别在指定区间范围上运行相应的 $\texttt{top}$ 查询，并检查 x 是否存储在输出块中或其下方。然后，我们根据范围计数结果 $\{w_i\}$ 确定扇出，进而确定输出块中的活动分隔键。对于每个非空的子节点引用，我们在辅助外存中分配一个块，并设置其父指针。最后，我们递归地重建活动区段各自区间内的子树，并将子树根节点的引用传入 T_X；该子树根节点包含重建其子树所需的所有键。请注意，对于这种递归形式，使用父节点指针即可实现重建过程，使其只需常数个内存存储临时块。子树在辅助外存中重建完成后，我们从 UR 中删除 m 个废弃块，并将 m' 个块移至 UR，从而得到最终树 T'。

显然，块写入的数量为$\mathcal{O}(m + m')$，块读取的数量为$\mathcal{O}(D' \cdot m)$，这可由简单的计费论证得出。也就是说，旧树中的 m 个块中的任意一个块 u，只有当 u 的键范围与该调用的查询范围（ℓ, r）相交时，才会被 $\texttt{top}$ 调用读取。因此，(ℓ, r) 包含非空 u 的最小键、最大键或全部键。此类读取中任意一种的数量均由 D' 界定，因为输出树中发起对 $\texttt{top}$ 调用的那些块，其区间依据搜索树性质彼此包含。我们的部分重建算法的预期 I/O 界将由下一节对随机二叉搜索树的深度和大小的分析得出。

3 随机二叉搜索树中的搜索界与子树权重

由于成功的树搜索会更早终止，因此只需分析对键 $q \notin X$ 的查询成本，即不成功搜索。在本节中，我们界定预期块读取次数

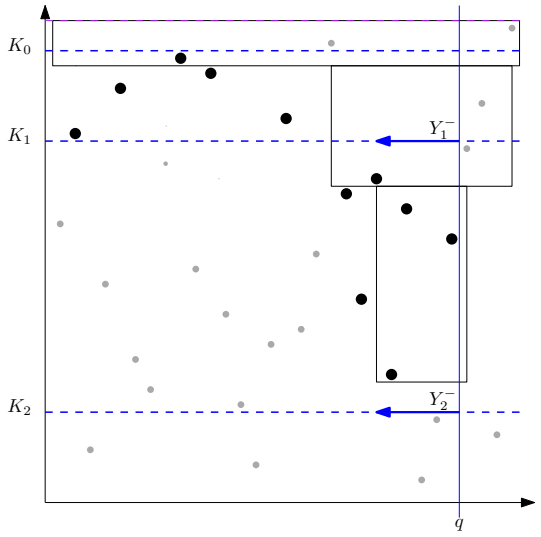

图 3 层划分 K_i 以及由 $Y_1^- = 5$ 和 统计的键数量的示意图 $Y_{\ \ 2}^- = 6$ （黑色）出现在引理 2 的证明中。

对于在块树 T_X 中搜索某个固定的 q，其中随机性来自排列 $\pi \in \mathrm{Perm}(n)$，即双射集合 $\pi\colon X \to \{1, \ldots, n\}$。

考虑随机二叉搜索树中从根节点到某个固定 q的搜索路径上的块序列 v_1, v_2, ...。由于子树权重是一个（严格）递减函数，我们可知节点的扇出 δ 是一个非递增函数，并最终取值为 ≤ 1。根据 δ 的定义，搜索路径上满足 $\delta = 1$ 的块在最坏情况下至多有 $O(1/\varepsilon)$ 个。因此，只需界定满足 $\delta \geq 2$ 的内部块的期望数量；这些块都具有搜索树性质 $[\ell(v_i)\ r(v_i)] \supset (\ell(v_{i+1})\ r(v_{i+1})) \ni q$。我们对随机二叉搜索树中范围搜索的块读取数量所作的界定将用到下一节的结果，因此会在该节末尾给出。

▶ **引理 2。** 设 X 为一个包含 n 个键的集合，$q \in \mathbb{R} \setminus X$，且 $2 \leq \alpha$ 为一个整数。随机变量 D_q 的期望值，即其区间中包含 q 的主树节点数量，至多为 $\mathbb{E}[D_q] \leq 5\log_\alpha n$。其区间中包含 q 的次级树节点数量（即 $\delta < \alpha + 1$）的期望值为 $O(1/\varepsilon)$。特别地，q 的搜索路径中的块的期望数量为 $O(\varepsilon^{-1} + \log_\alpha n)$。

随机二叉搜索树的块结构不允许将 Seidel 基于“祖先事件”对 Treap 的分析清晰地扩展 [18]或将几何搜索结构中点定位成本的反向分析扩展到 [8, 第 6 章。4]。我们的证明技术在某种程度上受随机化算法针对 静态 问题的自顶向下分析技术启发，该技术考虑“平均冲突列表”的大小。然而，要分析位置 q 由用户指定而非由算法随机选择的动态场景，还需要一些新的思想。我们的证明采用分层划分，各层大小以因子 α 呈指数增长，并对该划分某一层中的键使用双向计数过程。

证明。将 n 划分为形如 $[\alpha^i, \alpha^{i+1}$ 的区间，其中索引为 $0 \leq i \leq \mathrm{blog}_\alpha n\mathrm{c}$。例如，排列 π 会将键 $x \in X$ 分配给唯一的层索引，即 i，其中 $\pi(x) \in [\alpha^i, \alpha^{i+1})$。对于内部节点 v，令 $p^*(v) = \min\{\pi(x)\colon\ x$ 存储在 v 中$\}$，令 $p^\dagger(v)$ 为该集合上的最大值。注意，$p^*(v) + \alpha - 1 \leq p^\dagger v$，因为每个主节点包含

恰好 α 个键。因此， D_q 统计节点 v，这些节点要么同时具有 $p^*(v)$ $p^\dagger(v) \in [\alpha^i, \alpha^{i+1})$，要么 $p^\dagger(v)$ 的层索引大于 i。我们界定第一类节点的期望数量，因为在最坏情况下，第二类节点至多有 $\lfloor \log_\alpha n \rfloor$ 个。对于每个层索引 i，定义一个随机变量 V_i，用于统计该层中的树节点

$$V_i(\pi) = \left| \left\{ v \in T_X(\pi) \ : \ \ell(v) < q < r(v) \text{ and } p^*(v), p^\dagger(v) \in [\alpha^i, \alpha^{i+1}) \right\} \right| \ ,$$

我们有 $D_q \le \lfloor \log_\alpha n \rfloor + V$，其中 $V = \sum_{i \ge 0} V_i$ 是第一类块的总数。证明的其余部分将给出 V_i 的期望界。

令 $K_i = \{x \in X : \pi(x) < \alpha^{i+1}\}$ 对每个索引 i， $K_i^- = \{x \in K_i : x < q\}$，并且 $K_i^+ = K_i \setminus K_i^-$。因此， $K_i \subset K_{i+1}$，且 $|K_i| = \alpha^{i+1} - 1$。定义 Y_i^- 为从 K_i^- 开始、小于 q 但不包含在 K_{i-1} 中的连续键的数量。类似地，随机变量 Y_i^+ 统计大于 q 且 Y_i 的那些键：$= Y_i^- + Y_i^+$ 是 K_i 中连续键的数量，这些键的范围包含 q，但不包含来自 K_{i-1} 的元素。由于主树节点中的所有键都是活动分隔键，对于每个 $\pi \in \mathsf{Perm}(n)$，我们有 $V_i(\pi) \le Y_i(\pi)/\alpha$。

接下来，我们基于一系列二元事件来界定 $\mathbb{E}[Y_i^-]$：从 q 开始，我们按键序降序考察元素 $x \in K_i^-$，若 $x \in K_i \setminus K_{i-1}$，则计为一次“成功”，直到观察到一次失败（$x \in K_{i-1}$）。如果 K_i^- 中没有更多元素，则实验继续在 K_i^+ 上按键升序进行。定义 Z_i 为实验终止后的成功次数，则有 $Y_i^-(\pi) \le Z_i(\pi)$。观察到 j 次成功后出现失败的概率为 $\frac{|K_{i-1}|}{|K_i|-j}$，至少为 $p_i := \frac{|K_{i-1}|}{|K_i|}$，对所有 $j \ge 0$ 和 $i > 0$ 均成立。

因此，$\mathsf{Pr}[Z_i = j] \le \mathsf{Pr}[Z_i' = j+1]$，其中随机变量 $Z_i' \sim NB(1, p_i)$ 表示在失败概率为 p_i 的一系列独立伯努利试验中，获得一次失败所需的试验次数。由于 $\mathbb{E}[Z_i'] = 1/p_i$，我们有

$$\mathbb{E}[Y_i^-] \le \mathbb{E}[Z_i] < \mathbb{E}[Z_i'] = 1/p_i = \frac{\alpha^{i+1} - 1}{\alpha^i - 1} < \frac{\alpha}{1 - 1/\alpha^i} \ . \tag{1}$$

因此，对于所有 $\alpha \ge 2$，我们都有 $\mathbb{E}[V_i] \le \mathbb{E}[Y_i]/\alpha < \begin{matrix} 2 & \alpha \\ \alpha & 1 - 1/\alpha^i \end{matrix} \le 4$，这表明引理关于主树节点（具有 $\delta = \alpha + 1$）的结论成立。

由于搜索路径上扇出为 $\delta \le 1$ 的节点在最坏情况下有 $\mathcal{O}(1/\varepsilon)$ 个，因此还需对搜索路径上具有 $2 \le \delta(n') \le \alpha$ 的次级节点的期望数量给出界，其中 $n' := |X(v)|$ 是搜索路径上最顶层缓冲节点 v 的子树权重。对于任意固定的 n'，这个随机变量仅取决于 n' 个键的相对顺序，而这些顺序在 $\mathsf{Perm}(n')$ 中均匀分布。由于左右任一 $\delta(n')$ 区段中的键的期望数量为 $(n' - \alpha)/\delta(n')$，因此 q 区段中的键的期望数量有一个形如 $(n' - \alpha)/\delta(n') = \mathcal{O}(\alpha/\varepsilon)$ 的上界。由于该界对所有 n' 均成立，因此无条件地成立。由此，节点数量的引理 $\mathcal{O}(1/\varepsilon)$ 期望上界可由以下事实推出：所有次级节点（具有 $\delta \ge 2$）恰好存储 α 个键。◀

我们将在分析第 5 节中的结构变化时再次使用这一技巧。根据前面证明中的层划分，我们可以得到更新作用于子树时其期望权重的上界，形式为 $\sum_i \frac{\alpha^{i+1} - \alpha^i}{n} \frac{n - \alpha^i + 1}{\alpha^i - 1} = \mathcal{O}(\alpha \log_\alpha n)$。在本文余下部分，我们将推导出相关工具，以证明一次更新的期望结构变化（即块写入次数）与此上界相差至多一个 $(1/\alpha^2)$ 倍。

4 使用自顶向下分析对大小的界限

我们的分析将频繁使用以下刻画：由集合中优先级值最小的 α 个元素所诱导的键集合 X of$n$ 的划分。

▶**观察 3.** 在 $\pi \in \mathrm{Perm}(n)$ 中，由前 α 个 n 键所诱导的划分，与方程 $X_1+X_2+\cdots+X_{\alpha+1}=n-\alpha$ 的解之间存在双射，其中变量 X_i 为非负整数。这个双射意味着该方程的解出现的概率相同。

换句话说， X_i 是 T_X 根节点下第 i 个区段中的键数，其中$1 \le i \le \alpha+1$。因此， X_i 可以视为 X 中位于根节点所存储的第 i 个键和第（$i+1$）个键之间的连续键数。例如，在键为 $\{1,\ldots,10\}$ 且块大小为 $\alpha=3$ 的随机二叉搜索树中，包含键值 4、7、8 的根块可由赋值 $X_1=3,\ X_2=2,\ X_3=0,\ X_4=2$ 表示。

接下来，我们分析辅助结构的影响，因为不带缓冲区的随机二叉搜索树 ($\beta=0$) 的大小可能会非常大。例如，大小为 $n=2\alpha$ 的子树所需块的期望数量 $\mathbb{E}[S]$ 为

$$\begin{aligned}\mathbb{E}[S] &= 1+\Pr[X_1>0]+\cdots+\Pr[X_{\alpha+1}>0]\\ &= 1+(\alpha+1)\left(1-\binom{n-1}{\alpha-1}/\binom{n}{\alpha}\right)=1+(\alpha+1)(1-\alpha/n)=1+(\alpha+1)/2\quad .\end{aligned}$$

N请注意，在此示例中，缓冲仅用一个额外的块就存储了整个子树 .

4.1 负载因子为 $1-\varepsilon$ 的均匀随机树的缓冲

我们对 (α,ε)-随机二叉搜索树的期望大小（因而也包括负载因子）进行自顶向下分析时，经常会用到以下计数和指标交换论证。

▶ **引理 4** (交换)。 对于所有 $i,j \in \{1,\ldots,\alpha+1\}$，均有 $\Pr[X_i<c]=\Pr[X_j<c]$。

证明。 根据观察 3， $X_1+\cdots+X_{\alpha+1}=n-\alpha$ 的每个解出现的概率都相同。因此，我们可以通过统计满足 $X_i \ge c$ 的解的数量，并将其除以解的总数，来计算 $\Pr[X_i \ge c]$。于是

$$\Pr[X_i<c]=1-\Pr[X_i\ge c]=1-\frac{\binom{n-c}{\alpha}}{\binom{n}{\alpha}}=1-\Pr[X_j\ge c]=\Pr[X_j<c]\ ,$$

如前所述。 ◀

接下来，我们给出含有 n 个键、优先级从 Perm(n) 中所有排列构成的均匀分布中取值的 (α， ε)-随机二叉搜索树的大小上界。我们的分析关键依赖于这样一个基本事实：将 Perm(n) 限制在任意一个基数为 $n'<n$ 的键子集上，可得到 Perm(n') 上的均匀分布。随机变量 S_n 表示空间，即随机二叉搜索树在一个含有 n 个键的集合上使用的块数； F_n 表示满块的数量； $E_n := S_n - F_n$ 表示非满块的数量。接下来，我们给出一个概率界，描述给定区段（例如 k）包含一定范围内 X_k 个键这一事件。

▶ **引理5**。 对于任意 $1 \le k \le \alpha+1$ 和 $i<j$，我们有 $\sum_{x=i}^{j}\Pr[X_k=x]=\binom{n-i}{\alpha}-\underline{\binom{n-1-j}{\alpha}\binom{n}{\alpha}}$。

证明。我们有概率$[X_k \ge i]=\binom{n-i}{\alpha}/\binom{n}{\alpha}$ 以及概率$[X_k \ge j+1]=\binom{n-(j+1)}{\alpha}/\binom{n}{\alpha}$。 ◀

这些基本事实使我们能够计算以下表达式。

▶ **引理 6**。我们有 $\sum_{x=1}^{m}\Pr[X_k=x]x=\left(\binom{n}{\alpha+1}-\binom{n-m}{\alpha+1}-m\binom{n-1-m}{\alpha}\right)/\binom{n}{\alpha}$。

证明。通过简单计算，我们有

$$\begin{aligned}\sum_{x=1}^{m}\Pr[X_k=x]x&=\sum_{x=1}^{m}\Pr[X_k=x]+\sum_{x=2}^{m}\Pr[X_k=x]+\cdots+\sum_{x=m}^{m}\Pr[X_k=x]\\&=\frac{\binom{n-1}{\alpha}-\binom{n-1-m}{\alpha}}{\binom{n}{\alpha}}+\ldots+\frac{\binom{n-m}{\alpha}-\binom{n-1-m}{\alpha}}{\binom{n}{\alpha}}\\&=\sum_{i=1}^{m}\frac{\binom{n-i}{\alpha}}{\binom{n}{\alpha}}-m\frac{\binom{n-1-m}{\alpha}}{\binom{n}{\alpha}}\ .\end{aligned}\tag{2}$$

请注意，一个包含 $n+1$ 个键、块大小为 $\alpha+1$ 的随机二叉搜索树满足$\Pr[X_1=i]=\binom{n-i}{\alpha}\binom{n+1}{\alpha+1}$。因此，由引理5 可得

$$\sum_{i=1}^{m}\Pr[X_1=i]=\sum_{i=1}^{m}\frac{\binom{n-i}{\alpha}}{\binom{n+1}{\alpha+1}}=\frac{\binom{(n+1)-1}{\alpha+1}-\binom{(n+1)-1-m}{\alpha+1}}{\binom{n+1}{\alpha+1}}\tag{3}$$

$$\Rightarrow\sum_{i=1}^{m}\binom{n-i}{\alpha}=\binom{n}{\alpha+1}-\binom{n-m}{\alpha+1}\tag{4}$$

在求和式 (4) 中使用 (2)，即可证明该引理。 ◀

现在我们可以证明大小界了。

▶ **定理 7** （大小）。一个 n 键（α,ε）-随机二叉搜索树中非满块的期望数量 $\mathbb{E}[E_n]$ 至多为 $\max\{\varepsilon n/\alpha,1\}$。

我们的证明对 n进行归纳，基例由次级缓冲结构所占用的非满块数量上界给出（见附录 A）。

证明。观察 15 指出，对于所有 $E_n\le 1$，只要 n 满足 $\delta_n=1$，就有此结论。此外，定理 18表明，对于 $\mathbb{E}[E_n]\le 27\delta_n$，$n\le\beta+\alpha=(\alpha+1)\rho+\alpha$。为简便起见，我们将分别用 t 和 γ代替 $\rho+\alpha$ 和 $(\alpha+1)\rho+\alpha$。由定理 18 可得，对于$t<n\le\gamma$，

$$\begin{aligned}\mathbb{E}[E_n]&\le 27\left\lceil\frac{n-\alpha}{\rho}\right\rceil=27\left\lceil\frac{n-\alpha}{108\alpha/\varepsilon}\right\rceil\\&\le 27\left(\frac{n-\alpha}{108\alpha/\varepsilon}+1\right)\le 27\cdot 2\frac{n-\alpha}{108\alpha/\varepsilon}=\frac{n-\alpha}{2\alpha/\varepsilon}\ .\end{aligned}$$

最后一个不等式成立，是因为 $n>\rho+\alpha=108\alpha/\varepsilon+\alpha$ 且 $\frac{n-\alpha}{108\alpha/\varepsilon}>1$。

对于 $n>\gamma$，我们通过归纳法证明该定理。对于每个索引 $i\in\{1,\ldots,\alpha+1\}$，定义$Y_i$ 为根节点下第 i 个区段中的非满块数量（在一个含有n 个键的随机二叉搜索树中）。注意，Y_i 只取决于 X_i，也就是第 i 个区段中的键数量及其相对优先级。因此我们有

$$\mathbb{E}[Y_i]=\sum_{x_i=1}^{n-\alpha}\Pr[X_i=x_i]\cdot\mathbb{E}[E_{x_i}]$$

$$\leq \sum_{x_i=1}^{t} \Pr[X_i = x_i] \cdot 1 + \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \frac{x_i}{2\alpha/\varepsilon} + \sum_{x_i=\gamma+1}^{n-\alpha} \Pr[X_i = x_i] \frac{x_i}{\alpha/\varepsilon} \tag{5}$$

$$= \left(\sum_{x_i=1}^{t} \Pr[X_i = x_i] + \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \right) - \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i]$$
$$+ \left(\sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \frac{x_i}{2\alpha/\varepsilon} - \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \frac{x_i}{\alpha/\varepsilon} \right) - 2 \sum_{x_i=1}^{t} \Pr[X_i = x_i] \frac{x_i}{2\alpha/\varepsilon}$$
$$+ \left(\sum_{x_i=1}^{t} \Pr[X_i = x_i] \frac{x_i}{\alpha/\varepsilon} + \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \frac{x_i}{\alpha/\varepsilon} + \sum_{x_i=\gamma+1}^{n-\alpha} \Pr[X_i = x_i] \frac{x_i}{\alpha/\varepsilon} \right) \tag{6}$$

$$= \left(\sum_{x_i=1}^{\gamma} \Pr[X_i = x_i] - \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \frac{x_i}{2\alpha/\varepsilon} - \sum_{x_i=1}^{t} \Pr[X_i = x_i] \frac{x_i}{2\alpha/\varepsilon} \right)$$
$$+ \sum_{x_i=1}^{n-\alpha} \Pr[X_i = x_i] \frac{x_i}{\alpha/\varepsilon} - \sum_{x_i=1}^{t} \Pr[X_i = x_i] \frac{x_i}{2\alpha/\varepsilon} - \sum_{x_i=t+1}^{\gamma} \Pr[X_i = x_i] \tag{7}$$

$$\leq \sum_{x_i=1}^{\gamma} \Pr[X_i = x_i] \left(1 - \frac{x_i}{2\alpha/\varepsilon} \right) + \frac{\mathbb{E}[X_i]}{\alpha/\varepsilon} \tag{8}$$

利用引理 4，可知对每个区段，求和项的值都相同$i \in \{1, \ldots, \alpha+1\}$。由于对于 $n \geq \alpha$，根节点是满的，且不计入 E_n，因此我们有

$$\mathbb{E}[E_n] = \mathbb{E}[Y_1 + \ldots + Y_{\alpha+1}] \leq (1+\alpha) \sum_{x_1=1}^{\gamma} \Pr[X_1 = x_1] \left(1 - \frac{x_1}{2\alpha/\varepsilon} \right) + \frac{n-\alpha}{\alpha/\varepsilon} \,. \tag{9}$$

要证明该不等式，只需说明9式右侧至多为$\frac{\varepsilon n}{\alpha}$。当且仅当$(1+\alpha)\sum_{x_1=1}^{\gamma} \Pr[X_1 = x_1] \left(1 - \varepsilon \frac{x_1}{2\alpha}\right) \leq \varepsilon$时，这一结论成立。由于 $\varepsilon > 0$ 且$\alpha + 1 > 0$,一只需说明该和的值不为正。当且仅当

$$\sum_{x_1=1}^{\gamma} \Pr[X_1 = x_1] \leq \frac{\varepsilon}{2\alpha} \sum_{x_1=1}^{\gamma} \Pr[X_1 = x_1] x_1 \,. \tag{10}$$

在左侧使用引理5 ，在右侧使用引理6后，剩下的只需证明

$$\frac{2\alpha}{\varepsilon} \binom{n-1}{\alpha} - \frac{2\alpha}{\varepsilon} \binom{n-1-\gamma}{\alpha} \leq \binom{n}{\alpha+1} - \binom{n-\gamma}{\alpha+1} - \gamma \binom{n-1-\gamma}{\alpha} \,. \tag{11}$$

通过简单计算可知，对于所有 $n > \gamma$，该不等式均成立。（完整证明见附录B 中的引理19。）
◀

由于每个满块包含 α 个不同键，即 $F_n \leq n/\alpha$，我们证明了 $S_n \leq (1+\varepsilon)n/\alpha$。因此，对于负载因子 L，即已分配块中键的相对利用率，可得到以下下界。

▶ **推论 8** (负载因子)。 设 $\varepsilon \in (0,\ 1/2]$ 且 $n \geq \alpha + \beta$。在 n 个键上，一个(α, ε)-随机二叉搜索树所占用的块的期望数量至多为 $(1+\varepsilon)n/\alpha$，即期望负载因子至少为 $1-\varepsilon$。

证明。 负载因子是随机变量 $L = n/\alpha S_n$。由于 $\phi(x) = 1/x$ 是凸函数，根据詹森不等式可得 $1/\mathbb{E}[X] \leq \mathbb{E}[1/X]$。利用引理 7，我们有 $\mathbb{E}[L] \geq \frac{n}{\alpha} \frac{1}{1+\varepsilon}() n/\alpha = \frac{1}{1+\varepsilon} \geq 1-\varepsilon$。◀

结合引理2 和定理 7，我们得到了以下界限，用于报告随机二叉搜索树中范围搜索的所有结果。

▶ **推论 9** （范围搜索）。 设 $\varepsilon \in (0, 1/2]$. 在(α, ε)-随机二叉搜索树中报告范围搜索的所有结果，期望块读取次数至多为 $O(\varepsilon^{-1} + k/\alpha + \log_\alpha n)$，其中 k 是结果键的数量。

现在，我们可以给出 (α, ε)-随机二叉搜索树写入效率的界。

5 基于部分重建的动态更新界

在次级 UR 树结构的设计中采用权重成比例的扇出，相较于使用由 $O(\beta/\alpha)$ 个块组成的简单列表，具有一个优势：当更新导致子树必须在缓冲与非缓冲状态之间切换时，无需进行额外的重构工作（参见区段 2.1）。结合上一节的空间界，以及更新随机二叉搜索树的部分重建算法至多只会重建受影响节点下方 3 棵子树（与其扇出无关）这一观察，我们便能给出本节中关于期望结构变化的界。

▶ **定理 10**。 设 $\varepsilon \in (0, 1/2]$。 n键 (α, ε)-随机二叉搜索树的一次更新，其期望总结构变化量为 $O\left(\frac{1}{\varepsilon} + \log_\alpha^\alpha n\right)$，即对于 $\alpha = \Omega(\log(n)/\log\log(n))$，为 $O(1/\varepsilon)$。

证明。 只需对从随机二叉搜索树中删除一个键时总结构变化量的期望给出上界x，因为插入该键的情形会涉及相同数量的块。设 $X' = X \cup \{x\}$ 分别表示删除 x 之前和之后的键集合，其中 $|X'| = n$。对于要应用第 v 2.1 节部分重建算法的子树根节点，有三种情形，即 $\delta_v \le 1$、 $\delta_v \in [2, \alpha]$和 $\delta_v = \alpha + 1$。

对于 $\delta_v \le 1$，最坏情况下，缓冲区列表中被修改的块最多有 $\mathcal{O}(1/\varepsilon)$ 个。

对于 $\delta_v = \alpha + 1$， v 是主树中的一个节点，我们将优先级值 $\{1, \ldots, n\}$ 按照形如 $[\alpha^i, \alpha^{i+1})$ 的区间进行层划分，其中整数 $i < \log_\alpha n$。键 x i $[\pi\ x\ \in [\alpha^i, \alpha^{i+1}\ ] =$ 的优先级 $\frac{\alpha^{i+1} - \alpha^i}{n} \le \alpha^{i+1}/n$ i $K =$ $x' \in X$ $\pi\ x'\ < \alpha^i$ 落入该层区间的概率为 Pr ( ))。对于层，令 { : ( ) } 表示将优先级为 $\ge \alpha^i$ 的键划分开来的分隔键集合，令随机变量 $Y(\pi)$ 表示 $X \setminus K$ 中与 x 位于同一区段的键的数量。因此，对于所有排列 π， v 的子树权重至多为 $|X(v)| \le Y$。利用观察 3 的刻画，我们还可得出，从 $\alpha + 1$ 来看， v 的三个子树的总期望权重至多为 $\frac{3}{\alpha+1} Y$。显然，根据定义，$\mathbb{E}[Y] = \sum_{n'} n' \Pr[Y = n']$，并且通过直接计算可得 $\mathbb{E}[Y] = \mathrm{O}(\frac{n - \alpha^i}{\alpha^i}) = \mathrm{O}(n/\alpha^i)$。请注意，每个区段中的键数量 n' 与它们的相对次序无关，即 $(n')!$ 种次序的可能性相同。因此，定理 7 表明，包含 $\frac{3}{\alpha+1} n'$ 个键的随机二叉搜索树中的块的期望数量有一个 $\frac{1+\varepsilon}{\alpha}$ 因子界。因此，对于层 i 中的 x， v 的三个子树中的块的期望数量 m' 至多为

$$\mathbb{E}[\ m' \mid \pi(x) \in [\alpha^i, \alpha^{i+1})\ ] = \sum_{n'} \mathbb{E}[\ m' \mid \pi(x) \in [\alpha^i, \alpha^{i+1})\ ,\ Y = n'\ ] \Pr[Y = n'] \tag{12}$$

$$\le \sum_{n'} \frac{1+\varepsilon}{\alpha} \left(\frac{3}{\alpha+1} n' \right) \Pr[Y = n'] \tag{13}$$

$$= 3 \frac{1+\varepsilon}{\alpha(\alpha+1)} \mathbb{E}[Y] = \mathcal{O}(n/\alpha^{i+2})\ . \tag{14}$$

因此，块写入的期望数量至多为

$$\mathbb{E}[m'] = \sum_{i=0}^{\lfloor \log_\alpha n \rfloor} \mathbb{E}\left[\, m' \mid \pi(x) \in [\alpha^i, \alpha^{i+1})\right] \Pr\left[\pi(x) \in [\alpha^i, \alpha^{i+1})\right]$$
$$\leq \sum_i \mathcal{O}\left(\frac{n}{\alpha^{i+2}}\right) \frac{\alpha^{i+1}}{n} = \mathcal{O}\left(\frac{\log_\alpha n}{\alpha}\right) \quad .$$

对于 $\delta_v \in [2, \alpha]$， v 是次级树的一个节点，其子树权重为 n' 满足$n' = \Theta(\delta_v \alpha/\varepsilon)$。因此，公式（13）在这种情况下为 $\sum_{n'} \frac{1+\varepsilon}{\alpha} \frac{3}{\delta_v} \Theta(\delta_v \alpha/\varepsilon) \Pr[Y = n'] = \mathcal{O}(1/\varepsilon)$。◀

参考文献

1 Arne Andersson 和 Thomas Ottmann。唯一表示字典的新紧确界。SIAM计算杂志，24(5):1091–1101，1995。 doi:10.1137/S0097539792241102。2 Ricardo A. Baeza-Yates。b+ 树的期望行为。《信息学学报》，26(5):439–471，1989。 doi:10.1007/BF00289146。3 Rudolf Bayer 和 Edward M. McCreight。大型有序索引的组织与维护。《信息学学报》，1:173–189，1972。 doi:10.1007/BF00288683。4 Michael A. Bender、Martin Farach-Colton、Jeremy T. Fineman、Yonatan R. Fogel、Bradley C. Kuszmaul 和 Jelani Nelson。缓存无关流式 B 树。收录于 Phillip B. Gibbons 和 Christian Scheideler 编，《SPAA》，第 81–92 页。ACM，2007。URL: https://doi.org/10.1145/1248377.1248393。5 Guy E. Blelloch 和 Daniel Golovin。具有应用的强历史无关哈希。载于第 48 届计算机科学基础研讨会论文集（FOCS'07），第 272–282 页，2007。 doi:10.1109/FOCS.2007.36。6 Guy E. Blelloch、Daniel Golovin 和 Virginia Vassilevska。用于计算几何的唯一表示数据结构。载于 Joachim Gudmundsson 编，《算法理论—SWAT 2008》，第 11 届斯堪的纳维亚算法理论研讨会，瑞典哥德堡，2008 年 7 月 2–4 日，论文集，《计算机科学讲义》第 5124 卷，第 17–28 页。施普林格，2008。doi:10.1007/978-3-540-69903-3_4。7 Gerth Stølting Brodal 和 Rolf Fagerberg。外存字典的下界。载于第 14 届 ACM-SIAM 离散算法年会论文集（SODA'03），第 546–554 页，2003。URL: https://dl.acm.org/doi/10.5555/644108.644201。8 Mark de Berg、Otfried Cheong、Marc J. van Kreveld 和 Mark H. Overmars。《计算几何：算法与应用》，第三版。施普林格，2008。doi:10.1007/978-3-540-77974-2。9 Amr Elmasry、Mostafa Kahla、Fady Ahdy 和 Mahmoud Hashem。具有常数更新时间的红黑树。《信息学学报》，56(5):391–404，2019。doi:10.1007/s00236-019-00335-9。10 John Esmet、Michael A. Bender、Martin Farach-Colton 和 Bradley C. Kuszmaul。tokufs 流式文件系统。载于 HotStorage。USENIX 协会，2012。URL: https://www.usenix.org/conference/hotstorage12/workshop-program/presentation/esmet。11 Daniel Golovin。具有隐私应用的唯一表示数据结构。博士论文，卡内基梅隆大学，2008。12 Daniel Golovin。B-Treap：B 树的一种唯一表示替代方案。载于第 36 届自动机、语言与编程国际研讨会论文集（ICALP'09），第 487–499 页，2009。 doi:10.1007/978-3-642-02927-1_41。13 William Jannen、Jun Yuan、Yang Zhan、Amogh Akshintala、John Esmet、Yizheng Jiao、Ankur Mittal、Prashant Pandey、Phaneendra Reddy、Leif Walsh、Michael A. Bender、Martin Farach-Colton、Rob Johnson、Bradley C. Kuszmaul 和 Donald E. Porter。Betrfs：一种右优化的写优化文件系统。载于 FAST，第 301–315 页。USENIX 协会，2015。URL: https://www.usenix.org/conference/fast15/technical-sessions/presentation/jannen。

14 Donald E. Knuth。《计算机程序设计艺术》第三卷：排序与查找。Addison-Wesley，1973。15 Klaus Küspert。采用广义溢出技术的 B*树存储利用率。《信息学学报》，19:35–55，1983。doi:10.1007/BF00263927。16 Ketan Mulmuley。《计算几何：随机化算法导论》。Prentice Hall，1994。17 Moni Naor 和 Vanessa Teague。反持久性：历史无关数据结构。载于第 33 届计算理论研讨会论文集（STOC'01），第 492–501 页，2001。doi:10.1145/380752.380844。18 Raimund Seidel 和 Cecilia R. Aragon。随机化搜索树。Algorithmica，16(4/5):464–497，1996。doi:10.1007/BF01940876。19 Lawrence Snyder。关于唯一表示数据结构。载于第 18 届计算机科学基础年会（SFCS'77），第 142–146 页，1977。doi:10.1109/SFCS.1977.22。20 Rajamani Sundar 和 Robert Endre Tarjan。集合与序列的唯一二叉搜索树表示及相等性测试。SIAM计算杂志，23(1):24–44，1994。doi:10.1137/S0097539790189733。21 Jeffrey Scott Vitter。外存算法与数据结构。ACM计算调查，33(2):209–271，2001。doi:10.1145/384192.384193。22 Andrew Chi-Chih Yao。关于随机2-3树。《信息学学报》，9:159–170，1978。doi:10.1007/BF00289075。

A 归纳基例：缓冲树的期望大小

▶ **引理 11.** 设 $X_1, \ldots, X_m$ 为一些非负整数随机变量，其中 $X_1 + X_2 + \cdots + X_m = n$。对于整数 $n \geq 1$ 和 $t \leq n$，我们有如下尾部界

$$\left(1 - \frac{t}{n+1}\right)^{m-1} \quad \leq \quad \Pr[X_i \geq t] \quad \leq \quad \left(1 - \frac{t}{n+m-1}\right)^{m-1}$$

证明。

$$\begin{aligned}\Pr[X_i \geq t] &= \frac{\binom{n-t+m-1}{m-1}}{\binom{n+m-1}{m-1}} = \frac{(n-t+m-1)!\ (n-t)!\ (m-1)!}{(n+m-1)!\ n!\ (m-1)!} \\ &= \frac{n-t+m-1}{n+m-1}\ \frac{n-t+m-2}{n+m-2}\ \cdots\ \frac{n-t+m-(m-1)}{n+m-(m-1)} \\ &= \left(1 - \frac{t}{n+m-1}\right)\left(1 - \frac{t}{n+m-2}\right)\cdots\left(1 - \frac{t}{n+m-(m-1)}\right)\end{aligned}$$

由于对于 $1 \leq i \leq m-1$，$\frac{t}{n+m-1} \leq \frac{t}{n+m-i} \leq \frac{t}{n+m-(m-1)} = \frac{t}{n+1}$，该引理成立。 ◀

▶ **引理 12.** 对于 $0 < c \leq a \leq b$，我们有 $\frac{a-c}{b-c} \leq \frac{a}{b}$。

证明。 $\frac{a-c}{b-c} \leq \frac{a}{b} \iff ab - bc \leq ab - ac \iff a \leq b$。 ◀

对于大小为 $n \leq \beta + \alpha =$ 的子树：$(\alpha+1)\rho + \alpha$，扇出参数 $\delta_n = \min\{\alpha+1, \max\{1, \lceil \frac{n-\alpha}{\rho} \rceil\}\}$ 等于 $\max\{1, \lceil \frac{n-\alpha}{\rho} \rceil\}$。因此，对于 n 且满足 $2 \leq \delta_n \leq \alpha+1$ 的情况，有 $(\delta_n - 1)\rho + \alpha < n \leq \delta_n\rho + \alpha$。此外，对于 $k \in \{2, 3, \ldots, \delta_n\}$，$n > (k-1)\rho + \alpha$ 当且仅当 $\delta_n \geq k$。

▶ **引理 13**。 对于大小为 $n \leq \beta + \alpha$ 的子树，其中 $\delta_n \leq \alpha + 1$，且所有 $k \in \{2, 3, \ldots, \delta_n\}$，我们有

$$\Pr[\delta_{X_i} \geq k] = \Pr[X_i > (k-1)\rho + \alpha] \leq \left(1 - \frac{k-1}{\delta_n}\right)^{\delta_n - 1},$$

其中 X_i 是根节点下第 i 个区段中的键数。

证明。

$$\Pr[X_i > (k-1)\rho + \alpha] = \Pr[X_i \geq (k-1)\rho + \alpha + 1]$$

将引理 m、t和 n 中的变量 11 分别设为 δ_n、$(k-1)\rho + \alpha + 1$和 $n - \alpha$。我们有

$$\Pr[X_i \geq (k-1)\rho + \alpha + 1] \leq \left(1 - \frac{(k-1)\rho + \alpha + 1}{n - \alpha + \delta_n - 1}\right)^{\delta_n - 1}.$$

由于 $n \leq \delta_n \rho + \alpha$，我们有

$$\Pr[X_i > (k-1)\rho + \alpha] \leq \left(1 - \frac{(k-1)\rho + \alpha + 1}{\delta_n \rho + \alpha - \alpha + \delta_n - 1}\right)^{\delta_n - 1}$$

利用引理 12，我们从 $\left(\frac{k-1}{\delta_n \rho + \alpha - \alpha + \delta_n - 1}\right.$ $\left.\frac{\rho + \alpha + 1}{}\right)$ 的分子和分母中减去 $\delta_n - 1$。因此

$$\Pr[X_i > (k-1)\rho + \alpha] \leq \left(1 - \frac{(k-1)\rho + \alpha + 1 - (\delta_n - 1)}{\delta_n \rho}\right)^{\delta_n - 1}.$$

由于 δ_n 至多为 $\alpha + 1$，因此 $\alpha + 1 - (\delta_n - 1)$ 为正数，从而得到

$$\Pr[X_i > (k-1)\rho + \alpha] \leq \left(1 - \frac{(k-1)\rho}{\delta_n \rho}\right)^{\delta_n - 1} = \left(1 - \frac{k-1}{\delta_n}\right)^{\delta_n - 1},$$

如上所述。 ◀

我们将结合引理 13 和12 ，得到以下结果。

▶ **推论 14**。对于 $2 \leq k \leq \delta_n$，我们有

$$\Pr[\delta_{X_i} \geq k] = \Pr[X_i > (k-1)\rho + \alpha] \leq \left(1 - \frac{k-1-1}{\delta_n - 1}\right)^{\delta_n - 1} \leq e^{-(k-2)}.$$

▶ **观察 15**。 对于具有 $\delta_n = 1$ 的子树，该数据结构是一个简单列表，并且至多有一个非满块。

▶ **引理 16**。 对于具有 $\delta_n = 2$ 的子树，非满块的期望数量为 $\mathbb{E}[E_n] \leq 5$。

证明。 我们将设计一种算法，计算非满块数量的上界。根节点有 α 个键，其余键随机划分为两个区段，分别有X_1和 X_2个键，即 $X_1 + X_2 = n - \alpha$。对于 $i \in \{1, 2\}$，如果 $\delta_{X_i} = 1$，则至多有一个非满块，算法无需再在该区段中继续。如果 $\delta_{X_i} = 2$，为了观察非满块，需要再次划分这 $X_i - \alpha$ 个键。不可能同时有X_1和 X_2 满足 $\delta_{X_i} = 2$，因为这意味着 $n = \alpha + X_1 + X_2 \geq \alpha + 2(\rho + \alpha) > 2\rho + 3\alpha$，这与假设矛盾。因此，在每次迭代中，算法要么停止，要么继续处理

其中一个区段。最后一步中有两个区段，每个区段至多有一个非满块。因此，期望迭代次数加 1 是 $\mathbb{E}[E_n]$ 的上界。

显然，$\mathbb{E}[E_n]$ 确实是 n 的增函数，因此每次迭代从根节点丢失 α 个键会导致更少的非满块。为了获得一个较弱的上界，我们假设在每一步中，优先级最高的键将 $n-1$ 个剩余键（而不是$n-\alpha$ 个键）分割成两个新的区段。这些键还包括根节点的另外 $\alpha-1$ 个键。只需证明迭代次数的期望至多为 4。

在第 k 次迭代中，优先级最高的 k 个键将 n 个键分割成 k 个区段，其大小分别为 $Y_{(k)1}$、$Y_{(k)2}$、.....、$Y_{(k)k}$ 个键，其中 $Y_{(k)1}+Y_{(k)2}+\cdots+Y_{(k)k}=n-k$。如果任意一个 $Y_{(k)i}$ s 超过 $\rho+\alpha$，就会发生第 $k+1$ 轮。由于前 k 个键是均匀随机选取的，方程 $Y_{(k)1}+Y_{(k)2}+\cdots+Y_{(k)k}=n-k$ 的所有解具有相同的概率。接下来，我们将求出对于 $i\in\{1,\ldots,k\}$ 的 $\Pr[Y_{(k)i}>\rho+\alpha]$ 的上界。将引理 11 的参数 m、t 和 n 分别设为 k、$\rho+\alpha$ 和 $n-k$。因此

$$\begin{aligned}\Pr[Y_i^{(k)}>\rho+\alpha]&\le\left(1-\frac{\rho+\alpha+1}{n-k+k-1}\right)^{k-1}\\&\le\left(1-\frac{\rho+\alpha-1}{n-1}\right)^{k-1}.\end{aligned}$$

由于 n 至少为 $2\rho+\alpha$，我们有

$$\begin{aligned}\Pr[Y_i^{(k)}>\rho+\alpha]&\le\left(1-\frac{\rho+\alpha-1}{2\rho+\alpha-1}\right)^{k-1}\\&\le\left(1-\frac{\rho}{2\rho}\right)^{k-1}=2^{-(k-1)}.\end{aligned}$$

最后一个不等式是引理 12的一个应用。利用并集界，我们有

$$\Pr[\exists Y_i^{(k)}>\rho+\alpha]\le k/2^{k-1}.$$

定义 Z 为迭代次数，定义 Z_k 为一个指示随机变量，用于表示第 k次迭代是否发生。由此可得

$$\begin{aligned}\mathbb{E}[Z]&=\sum_{k=1}^{\infty}\mathbb{E}[Z_k]=\sum_{k=1}^{\infty}\Pr[Z_k=1]\\&\le\sum_{k=1}^{\infty}k/2^{k-1}=2(\sum_{j=1}^{\infty}\sum_{k=j}^{\infty}2^{-k})=2(\sum_{j=1}^{\infty}\frac{2^{-j}}{1-\frac{1}{2}})=2\frac{\frac{2^{-1}}{1-1/2}}{1/2}=4.\end{aligned}$$

◀

▶ **引理 17**。 对于一个满足 $\delta_n=3$ 的子树，其非满块的期望数量为 $\mathbb{E}[E_n]\le 10$.

证明。 $n-\alpha$ 根节点下方的键分为三个区段，分别为 X_1，X_2、以及 X_3个键。与引理 16类似，我们可以证明，不可能对每个 $\delta_{X_i}\ge 2$ 都成立$i\in\{1,2,3\}$，也不可能对某个 $\delta_{X_i}=\delta_{X_j}=3$ 成立 $i\ne j\in\{1,2,3\}$。由于随机变量的对称性 X_i，我们可以得出

$$\begin{aligned}1=\ &3\Pr[\delta_{X_1}=3\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1]\\&+3\Pr[\delta_{X_1}=3\wedge\delta_{X_2}=2\wedge\delta_{X_3}=1]\\&+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1]\end{aligned}$$

$$+\Pr[\delta_{X_1}=1\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1]\ .$$

接下来，我们对 $2\rho+\alpha<n\leq 3\rho+\alpha$ 进行归纳。根据观察 15、引理 16以及归纳假设，可得

$$\begin{aligned}
\mathbb{E}[E_n]\leq\ & 3\Pr[\delta_{X_1}=3\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1](10+1+1)\\
&+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=2\wedge\delta_{X_3}=1](5+5+1)\\
&+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1](5+1+1)\\
&+\Pr[\delta_{X_1}=1\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1](1+1+1)\\
<\ & 3\Pr[\delta_{X_1}=3\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1](12-7)\\
&+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=2\wedge\delta_{X_3}=1](11-7)\\
&+7(3\Pr[\delta_{X_1}=3\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1]\\
&+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=2\wedge\delta_{X_3}=1]\\
&+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1]\\
&+\Pr[\delta_{X_1}=1\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1])\\
=\ & 3\Pr[\delta_{X_1}=3\wedge\delta_{X_2}=1\wedge\delta_{X_3}=1]5+3\Pr[\delta_{X_1}=2\wedge\delta_{X_2}=2\wedge\delta_{X_3}=1]4+7\\
&\leq 7+3\Pr[\delta_{X_1}\geq 3]5+3\Pr[\delta_{X_1}\geq 2\wedge\delta_{X_2}\geq 2]4\ .
\end{aligned}$$

推论 14 意味着 $\Pr[\delta_{X_1}\geq 3]\leq\frac{1}{9}$。为完成证明，我们证明 $\Pr[\delta_{X_1}\geq 2\wedge\delta_{X_2}\geq 2]\leq\frac{1}{9}$。因此，我们得到所需的不等式 $\mathbb{E}[E_n]\leq 7+3\frac{5}{9}+3\frac{4}{9}=10$。

两个最高优先级键将这些键划分为三部分，分别包含 Y_1、Y_2和 Y_3 个键，其中 $Y_1+Y_2+Y_3=n-2$。第 i 部分包含根节点下方的所有 X_i 个键，以及根节点中的一些额外键，因此对于每个 $i\in\{1,2,3\}$，都有 $X_i\leq Y_i$。

$$\begin{aligned}
\Pr[\delta_{X_1}\geq 2\wedge\delta_{X_2}\geq 2]&=\Pr[X_1\geq\rho+\alpha+1\wedge X_2\geq\rho+\alpha+1]\\
&\leq\Pr[Y_1\geq\rho+\alpha+1\wedge Y_2\geq\rho+\alpha+1]\\
&=\frac{\binom{n-2(\rho+\alpha+1)}{2}}{\binom{n}{2}}
\end{aligned}$$

引理 11的相同技巧给出

$$\frac{\binom{n-2(\rho+\alpha+1)}{2}}{\binom{n}{2}}\leq\left(1-\frac{2(\rho+\alpha+1)}{n}\right)^2\ .$$

结合前面的不等式以及 $n\leq 3\rho+\alpha$，可得

$$\Pr[\delta_{X_1}\geq 2\wedge\delta_{X_2}\geq 2]\leq\left(1-\frac{2(\rho+\alpha+1)}{3\rho+\alpha}\right)^2\ .$$

从分子和分母中减去 α。根据引理 12，我们有：

$$\Pr[\delta_{X_1}\geq 2\wedge\delta_{X_2}\geq 2]\leq\left(1-\frac{2\rho+\alpha+2}{3\rho}\right)^2\leq\left(1-\frac{2\rho}{3\rho}\right)^2=\frac{1}{9}\ ,$$

如所要求。 ◀

▶ **定理 18.** 对于大小为 $n\leq\beta+\alpha$ 的子树，非满块的期望数量为 $\mathbb{E}[E_n]\leq 27\cdot\delta_n$。

证明。 我们将对 n进行归纳证明。观察 15、引理 16和引理17 给出了归纳的基础。我们通过使用随机变量 X_1、.....、 X_{δ_n}将问题划分为子问题。

$$\mathbb{E}[E_n] = \mathbb{E}[E_{X_1}] + \mathbb{E}[E_{X_2}] + \cdots + \mathbb{E}[E_{X_{\delta_n}}] = \delta_n \mathbb{E}[E_{X_1}]$$

注意， $X_i \le n - \alpha$，因此可以对其应用归纳假设。对于 $n \ge 4\rho + \alpha$，我们将使用归纳假设和基例，得到

$$\begin{aligned}
\mathbb{E}[E_{X_i}] &\le \Pr[\delta_{X_i} = 1] \cdot 1 + \Pr[\delta_{X_i} = 2] \cdot 5 + \Pr[\delta_{X_i} = 3] \cdot 10 + \sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} = k] \cdot (27 \cdot k) \\
&< \Pr[\delta_{X_i} = 1] \cdot 10 + \Pr[\delta_{X_i} = 2] \cdot 10 + \Pr[\delta_{X_i} = 3] \cdot 10 + \sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} = k] \cdot (27 \cdot k) \\
&= 10 \Pr[1 \le \delta_{X_i} \le 3] + 27 \sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} = k] \cdot k \\
&\le 10 + 27 \sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} = k] \cdot k
\end{aligned}$$

我们可以将 $\sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} = k] \cdot k$ 改写为 $3\Pr[\delta_{X_i} \ge 4] + \sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} \ge k]$。然后利用推论14，得到

$$\begin{aligned}
\mathbb{E}[E_{X_i}] &\le 10 + 27(3 \Pr[\delta_{X_i} \ge 4] + \sum_{k=4}^{\delta_n} \Pr[\delta_{X_i} \ge k]) \\
&\le 10 + 27 \left(3e^{-2} + \sum_{k=4}^{\infty} e^{-(k-2)} \right) \\
&\le 10 + 27 \left(\frac{3}{e^2} + \sum_{k=2}^{\infty} e^{-k} \right) \\
&= 10 + 27 \left(\frac{3}{e^2} + \frac{e^{-2}}{1 - e^{-1}} \right) \le 10 + 0.6202 \times 27 < 27 \ ,
\end{aligned}$$

如上所述。 ◀

B 补充证明

▶ **引理 19.** 考虑一个包含 $n > \gamma$ 个键的子树，其中 γ 等于 $\beta + \alpha = (\alpha + 1)\rho + \alpha$。我们有：

$$2\frac{\alpha}{\varepsilon}\binom{n-1}{\alpha-i} - 2\frac{\alpha}{\varepsilon}a\binom{n-1-\gamma}{\alpha-i} \le \binom{n}{\alpha+1-i} - \binom{n-\gamma}{\alpha+1-i} - \gamma\binom{n-1-\gamma}{\alpha-i} \ .$$

证明。 通过对 n进行归纳来证明。对于 $n = \gamma + 1$ ，该命题为

$$2\frac{\alpha}{\varepsilon}\binom{\gamma}{\alpha-i} \le \binom{\gamma+1}{\alpha+1-i} \tag{15}$$

$$\iff \quad 2\frac{\alpha}{\varepsilon}\frac{\gamma(\gamma-1)\dots(\gamma-\alpha+i+1)}{(\alpha-i)!} \le \frac{(\gamma+1)\gamma\dots(\gamma-\alpha+i+1)}{(\alpha-i+1)(\alpha-i)!} \tag{16}$$

$$\iff \quad 2\frac{\alpha}{\varepsilon}(\alpha-i+1) \le \gamma+1 \quad , \tag{17}$$

这是因为 $\gamma = (\alpha+1)\rho + \alpha = 108(\alpha+1)\alpha/\varepsilon + \alpha > 2(\alpha+1)\alpha/\varepsilon$。

假设对于每个 $\gamma < m \le n$，引理均成立。我们将证明引理对于 $n+1$ 成立，即我们将证明：

$$2\frac{\alpha}{\varepsilon}\binom{n}{\alpha-i} - 2\frac{\alpha}{\varepsilon}\binom{n+1-1-\gamma}{\alpha-i} \le \binom{n+1}{\alpha+1-i} - \binom{n+1-\gamma}{\alpha+1-i} - \gamma\binom{n+1-1-\gamma}{\alpha-i}$$

对于所有 i 均成立。由于不等式对于 $i = \alpha - 1$ 成立，因为

$$\begin{aligned} 2\frac{\alpha}{\varepsilon}n - 2\frac{\alpha}{\varepsilon}(n-\gamma) &\le \frac{n(n+1)}{2} - \frac{(n-\gamma)(n-\gamma+1)}{2} - \gamma(n-\gamma) \\ \iff \quad 2\frac{\alpha}{\varepsilon}\gamma &\le \frac{2\gamma n - \gamma^2 + \gamma}{2} - \frac{2\gamma n - 2\gamma^2}{2} \\ \iff \quad 2\frac{\alpha}{\varepsilon}\gamma &\le \frac{2n + \gamma^2 + \gamma}{2} \quad . \end{aligned}$$

右侧大于 $\frac{\gamma^2+\gamma}{2}$。因此，只需注意到——

$$2\frac{\alpha}{\varepsilon}\gamma \le \frac{\gamma^2+\gamma}{2} \quad \iff \quad 4\frac{\alpha}{\varepsilon} \le \gamma + 1 = (\alpha+1)\rho + \alpha + 1 = (\alpha+1)108\alpha/\varepsilon + \alpha + 1$$

这成立。

对于 $i < \alpha - 1$，我们将结合归纳假设使用恒等式 $\binom{m}{k} = \binom{m-1}{k} + \binom{m-1}{k-1}$。我们有

$$\begin{aligned} &2\frac{\alpha}{\varepsilon}\binom{n}{\alpha-i} - 2\frac{\alpha}{\varepsilon}\binom{n-\gamma}{\alpha-i} \\ &= 2\frac{\alpha}{\varepsilon}\binom{n-1}{\alpha-(i+1)} + 2\frac{\alpha}{\varepsilon}\binom{n-1}{\alpha-1} - 2\frac{\alpha}{\varepsilon}\binom{n-1-\gamma}{\alpha-(i+1)} - 2\frac{\alpha}{\varepsilon}\binom{n-1-\gamma}{\alpha-i} \\ &= \left[2\frac{\alpha}{\varepsilon}\binom{n-1}{\alpha-(i+1)} - 2\frac{\alpha}{\varepsilon}\binom{n-1-\gamma}{\alpha-(i+1)}\right] + \left[2\frac{\alpha}{\varepsilon}\binom{n-1}{\alpha-i} - 2\frac{\alpha}{\varepsilon}\binom{n-1-\gamma}{\alpha-i}\right] \\ &\le \binom{n}{\alpha+1-(i+1)} - \binom{n-\gamma}{\alpha+1-(i+1)} - \gamma\binom{n-1-\gamma}{\alpha-(i+1)} \\ &+ \binom{n}{\alpha+1-i} - \binom{n-\gamma}{\alpha+1-i} - \gamma\binom{n-1-\gamma}{\alpha-i} \\ &= \binom{n+1}{\alpha+1-i} - \binom{n+1-\gamma}{\alpha+1-i} - \gamma\binom{n-\gamma}{\alpha-i} \quad . \end{aligned}$$

◀