

```

IX:{Series |({} {2}
IX:{Series!geometric|({} {4}
IX:{Euler's constant} {4}
IX:{Series!geometric|)} {4}
IX:{Series!arithmetic|({} {4}
IX:{Series!arithmetic|)} {5}
IX:{Series!harmonic|({} {5}
IX:{Euler's constant} {5}
IX:{Series!harmonic|)} {5}
IX:{Series|)} {5}

```

图 4-74 练习 4.53 的样本输入

```

Euler's constant: 4,5
Series: 2-5
arithmetic: 4-5
geometric: 4
harmonic: 5

```

图 4-75 练习 4.53 的样本输出

参考文献

关于二叉查找树更多的信息，特别是树的数学性质可以在 Knuth[22]和[23]的两本书中找到。

有几篇论文讨论了由在二叉查找树中的有偏删除(biased deletion)算法引起的平衡不足问题。Hibbard 的论文[19]提出原始删除算法并确立了一次删除保持树的随机性的结论。文献[20]和[5]分别对只有 3 个节点的树和 4 个节点的树进行了全面的分析。Eppinger 的论文[14]提供了非随机性的早期经验性的证据，而 Calberson 和 Munro 的论文[10]和[11]则提供了某些解析论证(但不是对混杂插入和删除的一般情形的完整证明)。

AVL 树由 Adelson-Velskii 和 Landis[1]提出。AVL 树的模拟结果以及高度的不平衡允许最多到 k 的各种变化在[21]中讨论。AVL 树的删除算法可以在[23]中找到。在 AVL 树中平均搜索开销的分析是不完全的，但是，[24]中得到了某些结果。

文献[3]和[8]考虑了类似本书 4.5.1 节类型的自调整树。伸展树在[28]中作了描述。

B 树首先出现在[6]中。原始论文中所描述的实现方法允许数据存储在内部节点也能存储在树叶上。我们描述过的数据结构有时叫做 B^+ 树。[9]对不同类型的 B 树进行了综合分析。[17]报告了各种方案的经验性结果。2-3 树和 B 树的分析可以在[4]、[13]以及[32]中找到。

练习 4.17 看上去很难。一种解法可以在[15]中找到。练习 4.29 取自[31]。在练习 4.43 中描述的 B^+ 树的信息可以在[12]中找到。练习 4.47 取自文献[2]。练习 4.48 的解法使用 $2N - 6$ 次旋转，该解法在[29]中给出。按照练习 4.50 的方式使用的线索(threads)首先在[27]中提出。k-d 树是最早在[7]中提出来的，将在本书第 12 章进行讨论，它处理多维数据。

另外一些流行的平衡查找树是红黑树[18]和赋权平衡树[26]。在第 12 章可以找到更多的平衡树方案，此外也可以在著作[16]、[25]以及[30]中找到。

1. G. M. Adelson-Velskii and E. M. Landis, "An Algorithm for the Organization of Information," *Soviet. Mat. Doklady*, 3 (1962), 1259-1263.
2. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass., 1974.
3. B. Allen and J. I. Munro, "Self Organizing Search Trees," *Journal of the ACM*, 25 (1978), 526-535.
4. R. A. Baeza-Yates, "Expected Behaviour of B^+ -trees under Random Insertions," *Acta Informatica*, 26 (1989), 439-471.
5. R. A. Baeza-Yates, "A Trivial Algorithm Whose Analysis Isn't: A Continuation," *BIT*, 29