M 叉查找树退化到甚至是二叉查找树，因为那时我们又将无法摆脱 $\log N$ 次访问了。

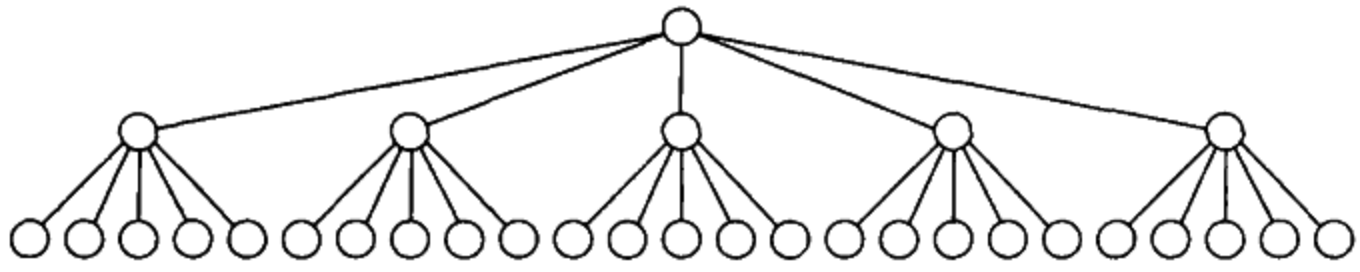

图 4-58　31 个节点的 5 叉树只有 3 层

实现这种想法的一种方法是使用 B 树。这里描述基本的 B 树[⊖]。许多的变种和改进都是可能的，但实现起来多少要复杂些，因为有相当多的情形需要考虑。不过，容易看到，原则上 B 树保证只有少数的磁盘访问。

阶为 M 的 B 树是一棵具有下列特性的树[⊖]：

1．数据项存储在树叶上。

2．非叶节点存储直到 $M-1$ 个关键字以指示搜索的方向；关键字 i 代表子树 $i+1$ 中的最小的关键字。

3．树的根或者是一片树叶，或者其儿子数在 2 和 M 之间。

4．除根外，所有非树叶节点的儿子数在$\lceil M/2 \rceil$和 M 之间。

5．所有的树叶都在相同的深度上并有$\lceil L/2 \rceil$和 L 之间个数据项，L 的确定稍后描述。

图 4-59 显示 5 阶 B 树的一个例子。注意，所有的非叶节点的儿子数都在 3 和 5 之间(从而有 2 到 4 个关键字)；根可能只有两个儿子。这里，我们有 $L=5$(在这个例子中 L 和 M 恰好是相同的，但这不是必须的)。由于 L 是 5，因此每片树叶有 3 到 5 个数据项。要求节点半满将保证 B 树不致退化成简单的二叉树。虽然存在改变该结构的各种 B 树的定义，但大部分在一些次要的细节上变化，而我们这个定义是流行的形式之一。

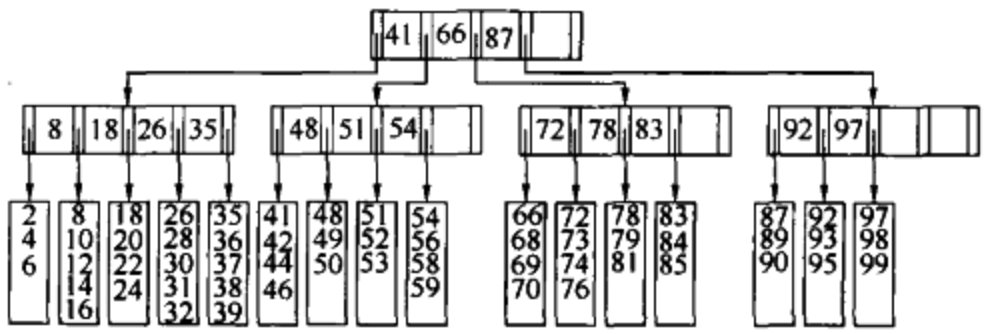

图 4-59　5 阶 B 树

每个节点代表一个磁盘区块，于是我们根据所存储的项的大小选择 M 和 L。例如，设一个区块能容纳 8192 字节。在上面的佛罗里达例子中，每个关键字使用 32 个字节。在一棵 M 阶 B 树中，有 $M-1$ 个关键字，总数为 $32M-32$ 字节，再加上 M 个分支。由于每个分支基本上都是另外的一些磁盘区块，因此可以假设一个分支是 4 个字节。这样，这些分支共用 $4M$ 个字节。一个非叶节点总的内存需求为 $36M-32$ 个字节。使得不超过 8192 字节的 M 的最大值是 228。因此，我们选择 $M=228$。由于每个数据记录是 256 字节，因此可以把 32 个记录装入一个区块中。于是，我们选择 $L=32$。这样就保证每片树叶有 16 到 32 个数据记录以及每个内部节点(除根外)至少以 114 种方式分叉。由于有 1 千万个记录，因此至多存在 625 000 片树叶。由此得知，在最坏情形下树叶将在第 4 层上。更具体地说，最坏情形的访问次数近似地由 $\log_{M/2} N$ 给出，这

⊖ 这里所描述的是通常称为 B^+ 树的树。

⊖ 法则 3 和 5 对于前 L 次插入必须要放宽。