b. 恰好存在$\lceil N/2 \rceil$片树叶。

c. 为找出它必须考查每一片树叶。

**6.9 证明，在一个大的完全堆(可以假设 $N=2^k-1$)中第 k 个最小元的期望深度以 $\log k$ 为界。

6.10 *a. 给出一个算法找出二叉堆中小于某个值 X 的所有节点。你的算法应该以 $O(K)$时间运行，其中，K 是输出的节点的个数。

b. 该算法可以扩展到本章讨论过的任何其他堆结构吗?

*c. 给出一个算法，最多使用大约 $3N/4$ 次比较找出二叉堆中任意的项 X。

*6.11 提出一个算法，以 $O(M+\log N \log\log N)$时间将 M 个节点插入到N 个元素的二叉堆中。证明该算法的时间界。

6.12 编写一个程序输入 N 个元素并

a. 将它们一个一个地插入到一个堆中。

b. 以线性时间建立一个堆。

比较这两个算法对于已排序、反序、以及随机输入的运行时间。

6.13 每个 deleteMin 操作在最坏情形下使用 $2\log N$ 次比较。

*a. 提出一种方案使得 deleteMin 操作只使用 $\log N+\log\log N+O(1)$次元素间的比较。这未必就意味着较少的数据移动。

**b. 扩展你在(a)部分中的方案使得只执行 $\log N+\log\log\log N+O(1)$次比较。

**c. 你能够把这种想法推向多远?

d. 在比较中节省下的资源能否补偿你的算法增加的复杂性?

6.14 如果一个 d-堆作为一个数组存储，对位于位置 i 的项，其父亲和儿子都在哪里?

6.15 设一个 d-堆初始时有 N 个元素，而我们需要对其执行 M 次 percolateUp 和 N 次 deleteMin。

a. 用 M、N 和d 表示的所有操作的总运行时间是多少?

b. 如果 $d=2$，所有的堆操作的运行时间是多少?

c. 如果 $d=\Theta(N)$，总运行时间是多少?

*d. 对 d 作什么选择将使总运行时间最小?

6.16 设二叉堆用显式链表示。给出一个简单算法来找出位于位置 i 上的树节点。

6.17 设二叉堆用显式链表示。考虑将二叉堆 lhs 和 rhs 合并的问题。假设这两个二叉堆均为满的完全树，分别包含 2^l-1 和 2^r-1 个节点。

a. 若 $l=r$，给出合并这两个堆的 $O(\log N)$算法。

b. 若$|l-r|=1$，给出合并这两个堆的 $O(\log N)$算法。

c. 给出合并这两个堆的与 l 和 r 无关的$O(\log^2 N)$算法。

6.18 最小-最大堆(min-max heap)是支持两种操作 deleteMin 和 deleteMax 的数据结构，每个操作用时 $O(\log N)$。该结构与二叉堆相同，不过，其堆序性质为：对于在偶数深度上的任意节点 X，存储在 X 上的元素小于它的父亲但是大于它的祖父(当这是有意义的时候)，对于奇数深度上的任意节点 X，存储在 X 上的元素大于它的父亲但是小于它的祖父，见图 6-57。

a. 如何找到最小元和最大元?

*b. 给出一个算法将一个新节点插入到该最小-最大堆中。

*c. 给出一个算法执行 deleteMin 和 deleteMax。

*d. 你能否以线性时间建立一个最小最大堆?