大字符数已知，那么该算法有可能节省一些时间。

上述两种方法相对来说都不难编码并可求解通常发表于杂志上的许多现实的字谜游戏。这些字谜通常有 16 行 16 列以及 40 个左右的单词。然而，假设我们把字谜变成为只给字谜板(puzzle board)而单词表基本上是一本英语词典，则上面提出的两种解法均需要相当长的时间来解决这个问题，从而这两种方法都是不可接受的。不过，这样的问题还是有可能在数秒内解决的，甚至单词表可以很大。

在许多问题当中，一个重要的观念是：写出一个工作程序并不够。如果这个程序在巨大的数据集上运行，那么运行时间就变成了重要的问题。我们将在本书看到对于大量的输入如何估计程序的运行时间，尤其是如何在尚未具体编码的情况下比较两个程序的运行时间。我们还将看到彻底改进程序速度以及确定程序瓶颈的方法。这些方法将使我们能够发现需要我们集中精力努力优化的那些代码段。

1.2 数学知识复习

本节列出一些需要记忆或是能够推导出的基本公式，并从推导过程复习基本的证明方法。

1.2.1 指数

$$X^A X^B = X^{A+B}$$

$$\frac{X^A}{X^B} = X^{A-B}$$

$$(X^A)^B = X^{AB}$$

$$X^N + X^N = 2X^N \neq X^{2N}$$

$$2^N + 2^N = 2^{N+1}$$

1.2.2 对数

在计算机科学中，除非有特别的声明，否则所有的对数都是以 2 为底的。

定义 1.1 $X^A = B$ 当且仅当 $\log_X B = A$。

由该定义可以得到几个方便的等式。

定理 1.1

$$\log_A B = \frac{\log_C B}{\log_C A};\ A,B,C>0, A\neq 1,\ C\neq 1$$

证明：

令 $X = \log_C B$，$Y = \log_C A$，以及 $Z = \log_A B$。此时由对数的定义，$C^X = B$，$C^Y = A$ 以及 $A^Z = B$，联合这三个等式则产生 $(C^Y)^Z = C^X = B$。因此，$X = YZ$，这意味着 $Z = X/Y$，定理得证。 ■

定理 1.2

$$\log AB = \log A + \log B;\ A,B>0$$

证明：

令 $X = \log B$，$Y = \log A$，以及 $Z = \log AB$。此时由于假设默认的底为 2，$2^X = A$，$2^Y = B$，及 $2^Z = AB$，联合最后的三个等式则有 $2^X 2^Y = 2^Z = AB$。因此 $X + Y = Z$，这就证明了该定理。 ■

其他一些有用的公式如下，它们都能够用类似的方法推导。

$$\log A/B = \log A - \log B$$

$$\log(A^B) = B\log A$$

$$\log X < X \text{ 对所有的 } X>0 \text{ 成立}$$

$$\log 1 = 0, \log 2 = 1, \log 1024 = 10, \log 1048576 = 20$$