

作为一个例子，图 2-12 中的程序段计算两个随机选取并小于或等于 N 的互异正整数互素的概率。（当 N 增大时，结果将趋向于 $6/\pi^2$ 。）

```
1      public static double probRelPrim( int n )
2      {
3          int rel = 0, tot = 0;
4
5          for( int i = 1; i <= n; i++ )
6              for( int j = i + 1; j <= n; j++ )
7                  {
8                      tot++;
9                      if( gcd( i, j ) == 1 )
10                         rel++;
11                  }
12
13          return (double) rel / tot;
14      }
```

图 2-12 估计两个随机数互素的概率

我们应该能够立即对这个程序做出分析。图 2-13 显示实际观察到的该例程在一台具体的计算机上的运行时间。该图表指出，表中的最后一列是最合适的，因此所得出的这个分析很可能正确。注意，在 $O(N^2)$ 和 $O(N^2\log N)$ 之间没有多大差别，因为对数增长得很慢。

N	CPU time(T)	T/N^2	T/N^3	$T/(N^2\log N)$
100	022	.002 200	.000 022 000	.000 477 7
200	056	.001 400	.000 007 000	.000 264 2
300	118	.001 311	.000 004 370	.000 229 9
400	207	.001 294	.000 003 234	.000 215 9
500	318	.001 272	.000 002 544	.000 204 7
600	466	.001 294	.000 002 157	.000 202 4
700	644	.001 314	.000 001 877	.000 200 6
800	846	.001 322	.000 001 652	.000 197 7
900	1 086	.001 341	.000 001 490	.000 197 1
1 000	1 362	.001 362	.000 001 362	.000 197 2
1 500	3 240	.001 440	.000 000 960	.000 196 9
2 000	5 949	.001 482	.000 000 740	.000 194 7
4 000	25 720	.001 608	.000 000 402	.000 193 8

图 2-13 对图 2-12 中例程的经验运行时间

2.4.6 分析结果的准确性

根据经验，有时分析会估计过大。如果这种情况发生，那么或者需要进一步细化分析（一般通过机敏的观察），或者可能是平均运行时间显著小于最坏情形的运行时间，不可能对所得的界再加以改进。对于许多复杂的算法，最坏的界通过某个坏的输入是可以达到的，但在实践中它通常是估计过大的。遗憾的是，对于大多数这类问题，平均情形的分析是极其复杂的（在许多情形下仍然悬而未决），而最坏情形的界尽管过分地悲观，但却是最好的已知解析结果。