挑战性题目DSCT501：大整数因子分解

问题描述：整数分解(integer factorization)又称素因数分解(prime factorization)，是将一个正整数写成若干素数的乘积。根据算术基本定理，这样的分解结是唯一的。整数分解问题在代数学、密码学、计算复杂性理论和量子计算机等领域中有重要意义。对一个输入的长整形正整数1 ~ 263进行因子分解。例如，输入5123478921，输出3\*17\*11\*11\*761\*1091。

思路分析[[1]](#footnote-1)与算法描述[[2]](#footnote-2)：

本人实现了基于素数筛的大整数质因子分解，并利用相关结论进行了优化处理。

由算数基本定理可知，任何正整数有且仅有一种分解为若干质数的积的形式。为了分解一个给定的大整数，我们需要先筛选出一定范围内的质数有哪些，再通过将质数作为除数对原数进行除法运算来判定其是否是大数的质因子。

首先筛选一定范围内的质数。选用欧拉筛，将从2开始的数的所有范围内的倍数划去后，下一次从未被划去的最小整数开始，并且只划去那些未被划去的数。时间复杂度为。

在筛选出一定范围内的质数并存储在数组Prime[]中后，若N mod Prime[i]==0，则Prime[i]是大数N的一个质因子；注意此处应不断循环，直到Prime[i]不再能整除N为止，++i。因为整数的质因子分解中允许重复出现相同的质因子。

接下来讨论优化方案。优化主要考虑上述Prime[]数组的大小，如何确定这个所需要的素数的范围是非常重要的，如果不进行优化那么我们将需要进行次的计算且需要的空间复杂度。考虑到如下的常用结论：

* 任意一个正整数N，最多只有一个质因数大于。

该结论的证明非常简单。设整数N有两个质因数，那么：



矛盾。故除了一个可能的质因数大于，其他的均在范围内。这就使得算法的时间与空间复杂度均降为了。

因此，算法的总体时间复杂度为。

[(1条消息) 快速幂 & 快速乘原理讲解（模板）\_LL\_Leung的博客-CSDN博客](https://blog.csdn.net/liangllhahaha/article/details/82119378)

[(1条消息) 素数//素数筛\_VUno)UKGzseI-CSDN博客\_素数筛](https://blog.csdn.net/dy416524/article/details/86431057)

[(1条消息) C++ 分解一个大数的素因数\_chenmeiqi的博客-CSDN博客](https://blog.csdn.net/qq_36770641/article/details/88853147)

1. 算法思路不但要给出解题的算法内涵，还应该分析该算法的时间复杂度。此页背面也可以答题，但不应续页。 [↑](#footnote-ref-1)
2. 代码的写作和测试建议使用GCC、G++等通用C/C++编译器进行编译以利于跨平台的性能测试。作业纸质版本和测试的可执行文件请提交给主管助教。测试样例命令格式为：DSCT501\_2018270103012.exe 5123478921。 [↑](#footnote-ref-2)