实验二　编程实现离散对数求解

实验内容

本周的任务是写一个程序来计算模素数p的离散对数。

令p是一个素数，g是有限乘法群$\mathbb{Z}_p^*$上的一个原根，然后给定一个$\mathbb{Z}_p^*$上的h满足$h = g^x$，其中$1 \leq x \leq 2^{40}$，目的是找到x。也就是说，你编写的程序以p, g, h作为输入，然后输出x。

该问题最直接的算法就是对x的2^{40}个可能的值逐个进行尝试，直到找到正确的一个，即直到找到一个x在$\mathbb{Z}_p$上满足$h = g^x$。这需要2^{40}次乘法运算。在本次实验中，你将要实现一个算法，该算法使用中间相遇攻击，时间代价约为$\sqrt{2^{40}} = 2^{20}$。

令$B = 2^{20}$。因为x是小于B^2，我们可以将未知的x写作$x = x_0 B + x_1$，其中$x_0, x_1 \in [0, B-1]$。然后，$h = g^x = g^{x_0 B + x_1} = (g^B)^{x_0} \cdot g^{x_1}$（在$\mathbb{Z}_p$上）。两边同时除以$g^{x_1}$，可得到$h/g^{x_1} = (g^B)^{x_0}$（在$\mathbb{Z}_p$上）。

上面等式中的变量是x_0和x_1，其他都是已知的：g和h是给定的，$b = 2^{20}$。由于x_0和x_1在等式的两边，所以我们可以使用中间相遇攻击来找到一个解：

- 为等式左边h/g^{x_1}的所有可能值创建一个哈希表，其中$x_1 = 0, 1, \cdots, 2^{20}$。
- 对于每一个$x_0 = 0, 1, \cdots, 2^{20}$，检查$(g^B)^{x_0}$是否在哈希表中，如果是，便找到了解$(x_0, x_1)$，即$x = x_0 B + x_1$。

总体工作大约是2^{20}次乘法来构建表，另外2^{20}次查找在此表中。

当你完成求解程序之后，请以附件**test.txt**中的p, g, h为输入，求解出x。

提示：本次实验需要使用支持多精度和模运算的环境。在Python中，可以使用**gmpy2**或**numbthy**模块。在C/C++中，可以使用**GMP**。

实验要求

- 请在线提交源码和实验报告。
- 实验报告需包括实验结果（x_0, x_1, x的值）、重要代码段解释以及本次实验总结。