再谈逆元

Ⅰ. 线性递推阶乘的逆元

一. 算法的数学依据：

首先，x 的逆元，我们可以理解成 1/x（Mod P）

所以就可得出以下公式：

$$\frac{1}{n!}=\frac{1}{(n+1)!}\cdot(n+1) \quad \text{(Mod P)}$$

二. 算法实现：

1. 在 O(n)的时间复杂度内，递推出 1~n 的阶乘
2. 使用费马小定理[1]，在 log（P）的时间复杂度内，求出 n!的逆元
3. 在 O(n)的时间复杂度内，倒序递推出 1~n 的阶乘的逆元

所以该算法的总体复杂度为 O(n)

三. 参考代码：http://paste.ubuntu.com/19081684/

Ⅱ. 线性递推 1~n 的逆元[2]

一. 算法的数学依据：

在使用上文的算法得到 n!的逆元后，不难得到下式

$$\frac{1}{n}=\frac{1}{n!}\cdot(n-1)! \quad \text{(Mod P)}$$

二. 算法实现：使用上文已经求得的阶乘逆元，线性计算即可

三. 代码参考：http://paste.ubuntu.com/19082411/

---

[1] 如果模数不是质数，那么递推时可能会出现有一些数没有逆元。所以一般该类题目的模数都是质数。

[2] 在之前的笔记中，已经提到一种线性的解法，但使用的数学公式不甚优雅，所以在此再记录一种非常优雅的算法。该算法在时间复杂度不变的前提下，让我们不用再手推复杂的公式