

伯努利数

这个数的定义来自于，之前人们对自然数幂的求解的过程中，出现的一种操作，感觉这个的主要目的在于求有关自然数幂和的问题上，所有的问题都不是直接给出的，多点见识。

$$\sum_{i=1}^n i^k = \frac{1}{k+1} * \sum_{i=1}^{k+1} C_{k+1}^i * B_{k+1-i} * (n+1)^i \quad B_0 = 1, \sum_{k=0}^n C_{n+1}^k * B_k = 0$$
$$B_n = -\frac{1}{n+1} \sum_{i=1}^{n-1} C_{n+1}^i * B_i$$

```
const int maxn = 2005;
ll C[maxn][maxn], b[maxn], Inv[maxn], tmp;
//需要预处理组合数，伯努力数，逆元，在 $o(n^2)$  的范围里解决这个问题
void init()
{
    C[0][0] = 1; //预处理组合数
    for(int i = 1; i < maxn; i++)
    {
        C[i][0] = 1;
        for(int j = 1; j <= i; j++) C[i][j] = (C[i-1][j-1] + C[i-1][j]) % mod;
    }
    Inv[1] = 1; //预处理逆元
    for(int i = 2; i < maxn; i++) Inv[i] = Inv[mod%i] * (mod - mod/i) % mod;
    b[0] = 1;
    for(int i = 1; i < maxn; i++) //预处理伯努力数
    {
        b[i] = 0;
        for(int k = 0; k < i; k++) b[i] = (b[i] + C[i+1][k] * b[k] % mod) % mod;
        b[i] = (b[i] * (-Inv[i+1])) % mod + mod % mod;
    }
}
ll slove(ll n, ll k) //处理的是前n项k次幂的情况
{
    n++; n %= mod; tmp = n;
    ll ans = 0;
    for(int i = 1; i <= k+1; i++)
    {
        ans = (ans + C[k+1][i] * b[k+1-i] % mod) * n % mod % mod;
        n = n * tmp % mod;
    }
    ans = ans * Inv[k+1] % mod;
    return ans;
}
```