逆元

人员

袁晨峻、杨俊彦、于跃、杨咏丞、刘奕辰、周治润、李锦澍、李雨谦、曹塬、许睿谦、隋天乙、王陆文龙 到课

作业

https://cppoj.kids123code.com/contest/753 (课上讲了 A ~ E 题, 课后作业是 F 题)

课堂表现

今天上课讲了 逆元 这个知识点, 同学们上课听讲都很认真。

逆元的内容其实结论非常简单非常好记, 同学们课下可以试着自己推一遍, 看看能不能自己把结论推出来

课堂内容

【模板】快速幂

logn 求快速幂 模板

```
#include <bits/stdc++.h>
using namespace std;
int a,b,p;
long long ans;
long long mi(long long y,long long x,int p)
{
    long long an=1;
    while(x)
    {
        if(x&1) an=an*y%p;
        y=y*y%p;
        x>>=1;
    }
    return an;
}
int main()
{
    cin>>a>>b>>p;
    ans=mi(a,b,p);
    cout<<a<<"^"<<b<<" mod "<<p<<"="<<ans;
    return 0;
}
```

【模板】模意义下的乘法逆元

当 mod 为质数时, a 的逆元就是 a 的 mod-2 次方, 除以 a 等价于 乘以 a 的逆元

```cpp
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
int mod;

int qmod(int a, int k) {
  int res = 1;
  while (k) {
    if (k&1) res = (LL)res * a % mod;
    a = (LL)a * a % mod;
    k >>= 1;
  }
  return res;
}

int inv(int x) { return qmod(x, mod-2); }

const int maxn = 3e6 + 5;
int fac[maxn], i_fac[maxn];

int main()
{
  int n; cin >> n >> mod;

  fac[0] = 1;
  for (int i = 1; i <= n; ++i) fac[i] = (LL)fac[i-1]*i % mod;

  i_fac[n] = inv(fac[n]);
  for (int i = n-1; i >= 0; --i) i_fac[i] = (LL)i_fac[i+1]*(i+1) % mod;

  for (int i = 1; i <= n; ++i) {
    cout << (LL)i_fac[i]*fac[i-1] % mod << "\n";
  }
  return 0;
}
```

【深基5.习7】杨辉三角

组合数递推, C[i][j] = C[i-1][j] + C[i-1][j-1]

```cpp
#include <bits/stdc++.h>

using namespace std;

const int maxn = 20 + 5;
int w[maxn][maxn];

int main()
```

```
{
  int n; cin >> n;
  for (int i = 0; i <= n-1; ++i) {
    w[i][0] = 1;
    cout << w[i][0];
    for (int j = 1; j <= i; ++j) {
      w[i][j] = w[i-1][j-1] + w[i-1][j];
      cout << " " << w[i][j];
    }
    cout << endl;
  }
  return 0;
}
```

组合数问题

C(n, m) = n! / (m! * (n-m)!)

O(n) 预处理 阶乘 和 阶乘的逆元, 后面就可以 O(1) 求 C(n,m) 了

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int mod = 998244353;

int qmod(int a, int k) {
  int res = 1;
  while (k) {
    if (k&1) res = (LL)res * a % mod;
    a = (LL)a * a % mod;
    k >>= 1;
  }
  return res;
}

int inv(int x) { return qmod(x, mod-2); }

const int maxn = 5e6 + 5;
int fac[maxn], inv_fac[maxn];

int C(int n, int m) {
  return (LL)fac[n] * inv_fac[m] % mod * inv_fac[n-m] % mod;
}

int main()
{
  ios::sync_with_stdio(false); cin.tie(0);

  int T, c; cin >> T >> c;
  fac[0] = 1; for (int i = 1; i <= c; ++i) fac[i] = (LL)fac[i-1]*i % mod;
  inv_fac[c] = inv(fac[c]); for (int i = c-1; i >= 0; --i) inv_fac[i] = (LL)inv_fac[i+1]*(i+1) % mod;

  int res = 0;
  while (T -- ) {
    int n, m; cin >> n >> m; res ^= C(n, m);
  }
  cout << res << endl;
  return 0;
}
```

【模板】模意义下的乘法逆元 2

用跟前面那个题类似的方法, 可以 O(n) 求 a1, a2, a3, ..., an 中所有数的逆元, 就可以 O(n) 做这个题了

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
int mod;

int qmod(int a, int k) {
  int res = 1;
  while (k) {
    if (k&1) res = (LL)res * a % mod;
    a = (LL)a * a % mod;
    k >>= 1;
  }
  return res;
}

int inv(int x) { return qmod(x, mod-2); }

const int maxn = 5e6 + 5;
int w[maxn], s[maxn], suf_s[maxn];

int read() {
  char ch = getchar();
  int res = 0, f = 1;
  while (!isdigit(ch)) {
    if (ch == '-') f = -1;
    ch = getchar();
  }
  while (isdigit(ch)) res = res*10 + (ch-'0'), ch = getchar();
  return res*f;
}

int main()
{
  int n, k; cin >> n >> mod >> k;

  s[0] = 1; for (int i = 1; i <= n; ++i) w[i] = read(), s[i] = (LL)s[i-1]*w[i] % mod;
  suf_s[n] = inv(s[n]); for (int i = n-1; i >= 0; --i) suf_s[i] = (LL)suf_s[i+1]*w[i+1] % mod;

  int res = 0, t = 1;
  for (int i = 1; i <= n; ++i) {
    t = (LL)t * k % mod;
    int value = (LL)t * suf_s[i] % mod * s[i-1] % mod;
    res = (res + value) % mod;
  }
  cout << res << endl;
  return 0;
}
```