组合数

```
struct CBN {
    vector<int> fac, invfac;
    i64 fsm(i64 x, int p){
        i64 ans = 1;
        i64 base = x;
        while(p){
            if(p & 1) (ans *= base) %= mod;
            (base *= base) %= mod;
            p >>= 1;
        }
        return ans;
    }
    CBN(int n){
        fac.resize(n + 1), invfac.resize(n + 1);
        fac[0] = 1;
        for(int i = 1;i <= n;i ++) fac[i] = 1ll * fac[i - 1] * i % mod;
        invfac[n] = fsm(fac[n], mod - 2);
        for(int i = n - 1;i >= 0;i --) invfac[i] = 1ll * (i + 1) * invfac[i + 1] % mod;
        return ;
    }
    int A(int n, int m){
        return 1ll * fac[m] * invfac[m - n] % mod;
    }
    int C(int n, int m){
        return 1ll * fac[m] * invfac[n] % mod * invfac[m - n] % mod;
    }
};
```

数据范围大

```
struct CBN {
    vector<vector<i64>> c;
    CBN(int n){
        c.resize(n + 1);
        for(int i = 0;i <= n;i ++) c[i].resize(n + 1);
        c[0][0] = 1;
        for(int i = 1;i <= n;i ++){
            c[0][i] = c[i][i] = 1;
            for(int j = 1;j < i;j ++){
                c[j][i] = c[j - 1][i - 1] + c[j][i - 1];
            }
        }
        return ;
    }
    int C(int n, int m){
        return c[n][m];
    }
};
```

数据范围小