伯努利数

- 定义与公式
  B_i定义为第i个伯努利数
  递归定义方式:
  B_0=1
  $$B_n=-\frac{1}{n+1}(C_{n+1}^0B_0+C_{n+1}^1B_1+...+C_{n+1}^{n-1}B_{n-1})$$
- 性质
  $$\sum_{k=0}^{n}C_{n+1}^kB_k=0$$
- 简单应用
- 自然数幂的前缀和
  $$\sum_{i=1}^{n}i^k=\frac{1}{k+1}\sum_{i=1}^{k+1}C_{k+1}^iB_{k+1-i}(n+1)^i$$

```
ll qpow(ll a,ll b){
        ll res=1;
        while(b){
            if(b&1) res=res*a%mod;
            b>>=1;
            a=a*a%mod;
        }
        return res;
}
ll c[maxn][maxn], inv[maxn], B[maxn];
void exgcd(ll a, ll b, ll &x, ll &y){
    if(b == 0){
        x = 1;y = 0;
        return ;
    }
    ll x1, y1;
    exgcd(b, a%b, x1, y1);
    x = y1;
     y = x1 - (a/b)*y1;
}
void get_fac(){
     for(int i=0; i<maxn; i++){
        c[i][0] = 1; c[i][i] = 1;
     }
     for(int i=1; i<maxn; i++)
        for(int j=1; j<=i; j++)
            c[i][j] = (c[i-1][j]+c[i-1][j-1])%mod;
}
void get_inv(){
     for(int i=1; i<maxn; i++){
         ll x, y;
         exgcd(i, mod, x, y);
```

```
            x = (x%mod+mod)%mod;
            inv[i] = x;
        }
}
void get_bb(){
        B[0] = 1;
        for(int i=1; i<maxn-1; i++){
            ll tmp = 0;
            for(int j=0; j<i; j++)
                tmp = (tmp+c[i+1][j]*B[j])%mod;
                B[i] = tmp;
                B[i] = B[i]*(-inv[i+1]);
                B[i] = (B[i]%mod+mod)%mod;
        }
}
ll get(ll n,int k){//返回(1^k+2^k+...n^k)%mod
        ++n;
        n %= mod;
        ll ans = 0;
        for(int i=1; i<=k+1; i++){
            ans = (ans+((c[k+1][i]*B[k+1-i])%mod)*qpow(n,(ll)i))%mod;
            ans = (ans%mod+mod)%mod;
        }
        ans = ans*inv[k+1];
        ans = (ans%mod+mod)%mod;
        return ans;
}
void init(){
        get_fac();
        get_inv();
        get_bb();
}
```

- FFT