快速幂

```
int quick_mid(int a,int b,int c) {
    int ans = 1;
    while (b) {
        if (b & 1) ans = ans * a % c;
        a = (a * a) % c;
        b >>= 1;
    }
    return ans;
}
```

快速乘法

```
int quick_mul(int a,int b, int c) {
    int ans = 0;
    while (b) {
        if (b & 1) ans = (ans + a) % c;
        a = (a + a) % c;
        b >>= 1;
    }
    return ans;
}
```

矩阵快速幂

```
#define maxn 1000
#define long long ll
#define MOD 1e9+7
#define mod(x) ((x) % MOD)
struct mat{
    int m[maxn][maxn];
}unit;
mat operator *(mat a, mat b) {
    mat ret;
    ll x;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            x = 0;
            for (int k = 0; k < n; k++) {
                x += mod((ll)a.m[i][k] * b.m[k][j]);
            }
            ret.m[i][j] = mod(x);
        }
    }
    return ret;
```

```
}
void init_unit() {
    for (int i = 0; i < maxn; i++) {
        unit.m[i][i] = 1;
    }
    return ;
}
mat pow_at(mat a, ll n) {
    mat ret = unit;
    while (n) {
        if (n & 1) ret = ret * a;
        a = a * a;
        b >>= 1;
    }
    return ret;
}
```