快速冪

```
//快速冪（通用版）
void fast_pow_rec(llint exp, llint* pow_table, bool* calculated, llint mod, llint*
ans)
{
    int p = log((double)exp) / log(2);
    llint remain = exp - pow(2, (double)p);

    if (calculated[p])
    {
        *ans *= pow_table[p];
        *ans %= mod;
    }
    else
    {
        for (int i = 1; i <= p; i++)
        {
            pow_table[i] = (pow_table[i - 1] * pow_table[i - 1]) % mod;
            calculated[i] = true;
        }
        *ans *= pow_table[p];
        *ans %= mod;
    }
    if (remain != 0)
        fast_pow_rec(remain, pow_table, calculated, mod, ans);
}

llint fast_pow(llint base, llint exp, llint mod)
{
    llint   pow_table[50];
    bool    calculated[50] = { 0 };

    pow_table[0] = base;    calculated[0] = true;
    llint ans = 1;
    fast_pow_rec(exp, pow_table, calculated, mod, &ans);
    return ans % mod;
}
```