题目链接:https://projecteuler.net/problem=207

题意

定义$4^t=2^t+k$是k的一个**拆分**，当且仅当$4^t,2^t,k$都是正整数，且t是实数。特别的当t也是正整数时，称为**完美拆分**。定义函数$P(m)$为$1\lt k \le m$中完美拆分占所有拆分的比例

题解

拆分的式子可以化简为$2^t(2^t-1)=k$不妨设$u=2^t$其中u是正整数，那么完美拆分显然就是u是2的整数次幂的情况，所以二分一下即可（应该是有单调性的吧

代码

```
#include<bits/stdc++.h>
#define ll long long
#define pii_ pair<int,int>
#define mp_ make_pair
#define pb push_back
#define fi first
#define se second
#define rep(i,a,b) for(int i=(a);i<=(b);i++)
#define show1(a) cout<<#a<<" = "<<a<<endl
#define show2(a,b) cout<<#a<<" = "<<a<<"; "<<#b<<" = "<<b<<endl
using namespace std;
const ll INF = 1LL<<60;
const int inf = 1<<30;
const int maxn = 2e5+5;
inline void fastio() {ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);}

bool check(ll k)
{
    ll l=2,r=1e9,t;
    while(l<=r){
        ll mid = (l+r)>>1;
        if(mid*(mid-1)<=k) l=mid+1,t=mid;
        else r=mid-1;
    }
    ll w = log2(t);
    return w*12345<t-1;
}

int main()
{
    fastio();
    ll L=1,R=1e17,ans;
    while(L<=R){
        ll mid = (L+R)>>1;
        if(check(mid)) ans=mid,R=mid-1;
```

```
        else L=mid+1;
    }
    cout<<ans<<endl;
    // Integer partition equations
    // 44043947822
    return 0;
}
```