

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 $1 \leq k \leq m$ 中完美拆分占有所有拆分的比例

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）

另外，直接枚举 2^t 中的 t 也是可以的

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;  
}
```

From:
<https://wiki.cvbbacm.com/> - **CVBB ACM Team**

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:wangzai_milk:wzx27:pe:207 

Last update: **2020/05/26 13:51**