1000 pt.

从二进制表示入手这一点是想到了，但是没有考虑清楚怎么将之前位上没有满足的剩余值放到下一层去递归。

需要明确一点，如果p位之前，那些没有满足的值，都是2^p的倍数，因此可以用一个r来表示。而这个r实际上范围是有限的，因为需要满足每种金币只有2个，也即2^p \* r<sum(2^p\*2+2^(p-1)\*2+…1\*2)，计算出来r<=3，如果r>3，都是0.

所以定义f(p,r)为已知p+1…63位这些个2的金币解决方案，被归纳为剩余2^p\*r需要余下的满足。余下的[0,p]这些金币的满足方案。且p范围是[-1,63]，而r的范围是[0,3]，都是相当小的，可以很容易计算。

用记忆化搜索或者DP都是可以的，这个也比较典型，需要明确计算的顺序。