杂题选讲

cly_none

Largest Prime

设 $P(k)$ 表示，最大的质数 p 使得存在 n 且满足 n^2+k^2 和 $(n+1)^2+k^2$ 同为 p 的倍数。

例如，$P(3)=13$，这是因为取 $n=6$，$n^2+k^2=39,(n+1)^2+k^2=52$，都是 13 的倍数。

求

$$\sum_{k=1}^{10\,000\,000} P(k)$$

答案有取模。

Source: Project Euler 659

Compromise or persist

Alice 在玩游戏。

一开始有 n 张牌，上面写着 1 到 n。在游戏的最开始这 n 张牌的顺序会被随机打乱。

每一轮，Alice 抽走一张牌，然而她不能看到牌上的数字。但是她的朋友 Bob 作为裁判可以看到数字，并告诉 Alice 已经抽到的牌里有多少张牌上面的数字比新抽上来的牌要小。

在至少抽一张牌后，Alice 随时可以结束游戏。或者在所有牌被抽光后游戏被迫结束。

Alice 会采取最优决策以最小化最后抽到的牌上的数字。你需要求出这个数字的期望值。

$n \leq 10^6$

Bonus: 证明 $n+1$ 的答案总是大于 n 的答案。

Compromise or persist

容易证明，当我们已经抽过 k 张卡牌后，我们的决策与之前卡片的相对排名无关。因此，我们可以将策略描述为一个数列 s_i，表示已经抽了 i 张牌后，最大的让我们不继续抽下去的 $i+1$ 的相对排名。很容易从后往前递推 s_i 得到答案。

不过我们可能会发现一个性质：$n+1$ 的答案总比 n 的答案要大。下面是一个有趣的证明。

我们称有 $n+1$ 牌的游戏为 G_{n+1}，有 n 张牌的游戏为 G_n。
考虑修改一下总共有 $n+1$ 张牌的游戏规则：当你抽到 $n+1$ 时，你会被立刻告知你抽到了 $n+1$。记这个新游戏是 T。

不难发现 T 的游戏结果一定优于 G_{n+1} 的游戏结果（白嫖了额外信息，避免了最后抽到 $n+1$），而 T 其实和 G_n 是一样的：你遇到 $n+1$ 一定直接跳过，若走到最后一个没有遇到 $n+1$ 则一定停下来。

因此 G_n 的游戏结果优于 G_{n+1}。

Subset Sums

我们要从集合 $\{1, 2, ..., 3p\}$ 中取出 p 个数，且满足这些数的和是 p 的倍数。
p 是一个质数。
求方案数，对 $1e9 + 9$ 取模。

Source: Project Euler 635 （有修改）

Divisors of $2n^2$

设 $f(n)$ 表示 $2n^2$ 不超过 n 的因数个数。

求

$$\sum_{n=1}^{10^{12}} f(n)$$

Source: Project Euler 735

Not Zeckendorf

考虑斐波那契数列 $\{1, 2, 3, 5, 8, 13, 21, \ldots\}$.

设 $f(n)$ 表示将 $n \geq 0$ 表示为**不同**的斐波那契数之和的方式。

例如，$16 = 3 + 13 = 1 + 2 + 13 = 3 + 5 + 8 = 1 + 2 + 5 + 8$，故 $f(16) = 4$.

求

$$\sum_{k=1}^{10^{13}} f(k)$$

Source: Project Euler 755 （有修改）

Not Zeckendorf

Quote Report 2

[redacted] Java Sun, 18 Apr 2021, 18:46

Interesting problem.

Let's list all fibonacci numbers upto N (for N = 10^13, there are L = 63 of them). Now, since we care only about representing some value X as sum of distinct fibonacci numbers, each representation of X can be written as a binary string with L bits.

Broadly speaking, if we consider all binary strings with L bits, each one of them correspond to one representation of a value. Now we want to count the number of binary strings, for which the sum doesn't exceed N.

Though doing naively is proportional to finding all representations naively.
Let's build a perfect binary tree with depth L. Then each leaf in this tree correspond to a mask, LR sequence obtained to reach this leaf from root.
Claim: There are very few nodes in this subtree for which leaves exist with sum <= N and sum > N.
Proof: I'd like to know.
Intuition: fibonacci values are quite large, and if two consecutive 1s in Zeckendorf's representation are decomposed, the sum is unattainable.

Hence, most nodes will have either all leaves in their subtree having sum <= N, therefore all count as ways to represent numbers upto N, or all leaves shall have sum > N, therefore none would count towards ways to represent numbers upto N. The remaining nodes are processed in branch and bound manner.

My Java code runs in 118ms.

Turán's water heating system

有 n 个零件，并且我们知道里面恰好有 m 个零件是好的，剩下的都是坏的。零件可以区别，但我们不知道哪些零件是好的。

某装置需要恰好两个好的零件。我们每次可以使用两个零件进行一次尝试，尝试成功当且仅当两个零件都是好的。（尝试失败零件不会被消耗）你能立刻知道尝试结果。

现在需要求出，在采取最优策略的情况下，至少尝试几次才能保证成功一次。

$n, m \leq 1e7$

如果你想要答案，直接搜索 Turán's Theorem 即可。

Source: Project Euler 713

Thank you