NOIP2016 模拟赛

Newnode

October 7, 2016

1 osiris

如果a比x大，我们就有了(a-x)/2的空余，否则就有x-a的需求，只要看需求是否比空余多就好了。

时间复杂度：O(1)。

2 obelisk

2.1 60%算法

这种题肯定是状压了，考虑如何压缩状态来保证不重不漏计数有向无环图。

我们给有向无环图的节点分层，第一层就是度为0的节点，第二层是删去第一层节点以后，剩下的度为0的节点，依次类推。

那么我们就可以记录$f_{i,j}$表示当前选择的节点状态是i，其中最后一层的节点状态是j。那么枚举下一层的节点状态就好，要求满足这下一层的节点每一个至少都和j中的节点有边，和其余的点的边则随意。

时间复杂度：$O(4^n m)$。

2.2 100%算法

试着将上一个算法优化一下，我们不记录j可以吗？

如果单纯的不记录j是会有重复的，因为一个点可能会被算到下面的层次上去。这个时候就可以容斥了。

令f_i表示当前选择的节点状态是i，枚举下一层状态k，那么只要乘上容斥系数$(-1)^{|k|+1}$即可。

时间复杂度：$O(3^n m)$，稍微优化即可至$O(3^n+2^n m)$。

3 ra

3.1 60%算法

我们要求$\sum_{1\le a,b\le n}[\mathtt{lcm(a,b)>n}]$，即求$\sum_{1\le a,b\le n}[\mathtt{lcm(a,b)} \le$ n]，显然后者更容易求。

考虑枚举a和b的最大公约数d，那么

$$\sum_{1\le a,b\le n}[\mathtt{lcm(a,b)} \le \mathtt{n}] = \sum_{1\le d\le n}\sum_{1\le a,b\le n/d}[\mathtt{ab} \le \mathtt{n/d}][\mathtt{gcd(a,b)=1}]$$

显然n/d只有O($\sqrt{n}$)种，于是可以枚举，那么考虑求

$$\sum_{1\le a,b\le k}[\mathtt{ab} \le \mathtt{k}][\mathtt{gcd(a,b)=1}]$$

可以假设a<b，那么$a \le \sqrt{k}$，那么$b \le k/a$且gcd(a,b)=1，这样的b的数量很好计算，直接容斥即为

$$\sum_{t|a}\mu(t)\lfloor k/at\rfloor - \varphi(a)$$

那么对于一个k，计算的复杂度为O($\sqrt{k}$log k)。

时间复杂度：直接这样计算，用积分计算复杂度为O($n^{0.75}$log n)。

3.2 80%算法

依然考虑求

$$\sum_{1\le a,b\le k}[\mathtt{ab} \le \mathtt{k}][\mathtt{gcd(a,b)=1}]$$

直接枚举a和b，那么对数为O(klog k)，那么对于$k \le n^{2/3}$的k，直接通过枚举a 和b用前缀和算出，剩下的依然用原来的办法。

时间复杂度：用积分计算复杂度为O($n^{2/3}$log n)。

3.3 100%算法

依然考虑求

$$f_k = \sum_{1\le a,b\le k}[\mathtt{ab} \le \mathtt{k}][\mathtt{gcd(a,b)=1}]$$

先求出

$$S_k=\sum_{1\le a,b\le k}[ab\le k]$$

可以发现枚举gcd(a,b)=t，有

$$f_k=S_k-\sum_{1<t}f_{k/t^2}$$

那么对于$k\ge n^{2/3}$的k就用这种方式迭代，复杂度为$O(\sqrt{k})$，而剩下可以用线性筛来计算。

时间复杂度：用积分计算复杂度为$O(n^{2/3})$。