2018 CCPC 网络赛题解

1001

将 n 个数拆成左右两列，左边表示买，右边表示卖，从左向右连边即表示买入和卖出。考虑第 i 天是否卖出，一定是在左边列的前 i - 1 个中找一个还未配对的最小值和其配对进行买卖获益最大，如果最小值 >= 当前第 i 天的价格就不交易。
配对时需要注意，如果当前配对的买入价格和之前某次配对的卖出价格相同，那么我们就可以将两条边合并为 1 条，交易次数减少而获利不变。
用堆维护可买入的东西，set / hash 维护已卖出的东西即可。

1002

首先根据裴蜀定理,当且仅当$gcd(C_1, C_2, \dots C_k, m) = 1$时有解
令g为所有非-1项的A_i的gcd,若$g > 0$,则可以枚举g的因子容斥
若$g = 0$,可以发现$f(m)$是一个类似欧拉函数的积性函数,转为求积性函数前n项和

1003

这里 p 是一个质数，由费马小定理，对于 $a \in \mathbb{Z}$

$$a^p \equiv a \mod p$$

所以对于 $0 \le x, y < p$,

$$(x+y)^p \equiv x + y \equiv x^p + y^p \mod p$$

于是只需要将加法与乘法定义为：

$$m + n := (m + n) \mod p$$

$$m \cdot n := (m \cdot n) \mod p$$

即可。至于集合相等的那个约束，验证一下可以发现是正确的。

1004

根据费马大定理可知n>2无解
n=0易知无解
其他情况根据费马大定理奇偶数列法则求解即可

1005

题意:
给定n, A, B,求

$$\sum_{i=0,i为奇数}^{n} \binom{n}{i} \sqrt{B}^i A^{n-i}$$

要求输出形如$\sum\limits_{i=1}^{q} a_i \sqrt{b_i}$的解.

解法:

$$\begin{aligned}\sum_{i=0,i为奇数}^{n} \binom{n}{i} \sqrt{B}^i A^{n-i} &= \frac{1}{2} \sum_{i=0}^{n} \binom{n}{i} \left(\sqrt{B}^i - (-\sqrt{B})^i\right) A^{n-i} \\ &= \frac{1}{2}\left(\left(A+\sqrt{B}\right)^n - \left(A-\sqrt{B}\right)^n\right)\end{aligned}$$

如何处理$\left(A+\sqrt{B}\right)^n - \left(A-\sqrt{B}\right)^n$?
考虑$(a+b\sqrt{B})(c+d\sqrt{B}) = (ac+bdB)+(bc+ad)\sqrt{B}$
用类似快速幂的分治算法即可求得
此外，由原始公式不难看出输出只会有一项，即 q 始终为 1

1006

$ans = 2\sum_{i=1}^{\Omega(2n)} c_i(2n)[c_i(2n)+c_{i+1}(2n)]$
其中$c_i(x)$是把x拆成i大于1的数乘积的方案数,可由$d_i(x)$(把x拆成i个数乘积的方案数)容斥出来,
即$c_i(x) = \sum_{j=0}^{i}(-1)^{i-j}C_i^j d_j(x)$
其中$d_i(x) = \prod_{k=1} C_{a_k+i-1}^{i-1}$, a_k是第k个质因子的指数
所以可以通过fft求得所有的$c_i(2n)$
因为满足$\sum a_k \le 1e5$,所以不同的取值只有$\sqrt{1e5}$个,所以预处理均摊复杂度$n\sqrt{n}$,fft复杂度$nlogn$,总复杂度$n\sqrt{n}$

1007

把循环节扒出来,把m归约到循环节长度大小,然后跑长度小于某个值的最长子段和就可以了

1008

ans=c(n, 4) - $\sum c(deg, 2) * (n-3) * 4$
用网络流跑这个试子

1009

对每条边单独计算贡献，一条边E将树分成两侧，假设其中一侧大小为M，则另一侧大小为$N-M$。
在$N!$条路线中每条都分为$N-1$段，对每段单独计算贡献，例如某一段从X到Y，则该段经过E当且仅当X与Y在E的两侧，对应的排列数为$2M(N-M)(N-2)!$。
共有$N-1$段，假设E的长度为L，则E的贡献为$2LM(N-M)(N-1)!$。

1010

先离散化到1e5 * 1e5 的格点，dp[i][j]表示走到(i,j)为止的得到的最大金钱，显然dp[i][j]=max{ dp[i-1][j],dp[i][j-1],dp[i-1][j-1]+v[i][j] }，v[i][j]表示斜着走可获得的金钱。接下来优化这个dp。显然最大值会在一个村庄取到，把带有权值的村庄(i，j)按照从上到下，从右到左的顺序更新（和01背包从右往左更新原因相同）。维护f[j]，f[j]表示走到第j列取得的金钱的最大值，且只在村庄处更新，所以f[j] = max(f[j] , max(f[0~j-1])+v[i][j])。最后答案取max{ f[0~MAX(j)] },MAX(j)表示离散化后的j的最大值
dp转移转化为区间最值查询问题，用树状数组或线段树维护即可。复杂度O(n*logn)