A

在j固定的情况下，c_i，$a_j+b_j\times c_i$的值越大。
故可以先对c_i排序，然后二分第k大的值S，然后求$\sum_{1\le i,j\le n}[a_j+b_j\times c_i\le S\}$
现在可以对每个j二分出有多少个i使得$a_j+b_j\times c_i\le S$，故可以在$O(n\lg n)$的时间内求出小于等于S的二元组个数。
可以通过subtask2，期望得分：100

B

题目大意：求$(((N^2+1)\mod a)+b)\mod c=N_3$的三元组$(a,b,c)$
算法1：枚举所有的a,b,c，复杂度$\Theta(P^3)$，可以过subtask1，期望得分30。

算法2：可以逆推得到N_2的所有取值及对应的c，然后可以得到所有N_1的取值及对应的a，对每个N_1的取值可以二分出所有合法的N_2的个数及求对应的方案。
复杂度$\Theta(P^2\lg P)$，可以过subtask1,2，期望得分65。

算法3：$\lfloor N_2/c\rfloor\times c+N_3=N_2$，发现求$N_1$的取值同算法2。
$\sum_{c=1}^{P}\lfloor P/c\rfloor=O(P\lg P)$，因此可以枚举$c$的值和$\lfloor N_2/c\rfloor$的值来得到$N_2$，且只有$O(P\lg P)$中取值。
然后可以用二分或者前缀和来得到合法的b。时间复杂度$O(P\lg P)$
期望得分100。

C

假设已知最大字典序情形的答案k，按顺序考虑每一位，贪心的让字典序尽量大。假设到了*第i回合*，为了不影响k，能出的牌一定是一个区间$[l, r]$，那么一定是尽量填一个最大的，考虑二分最大值x，*每次*希望判断在第i次出x会*不*会影响答案。
考虑对值维护一个线段树，每个节点记录一个三元组(S, L, R)，其中S*表示*当前值域的得分，L*和*R*分别*表示青蛙和柯南在当前区间还剩下多少牌。
那么合并信息时，在这个区间内青蛙将贪心的获得$d = \min\left(L_{rightchild}, R_{leftchild}\right)$的分数，具体策略为当柯南用左孩子中的$d$*张*牌时，使用右孩子代表的值域中的$d$张牌击败他。并将剩余的值上传。

D

首先二分答案，问题即转化为：是否可能 m 天后竹子的高度都不超过 X?

把砍竹子和竹子生长的过程反过来，反过程形如：一开始竹子的高度不超过 X. 每天早上每根竹子缩短 a_i 长度，如果缩短到负数，就失败；否则晚上可以在所有竹子中选 k 根加上 p 长度。m 天后，竹子的高度至少都要为 h_i，求是否有可能获胜。

这样一来，问题就简单了。每天有 k 次加长机会，如果有竹子将在接下来一天早上缩短到负数，就取出这棵竹子加长，如果没有机会就失败；如果机会用不完可以保留下来留给以后。最后第 m 天，若一根竹子高度为 $h'_i < h_i$，就要消耗 $\left\lceil\frac{h_i - h'_i}{p}\right\rceil$ 次此前留下的机会补足 h_i 的高度。用堆（我写的是非递归线段树）维护每根竹子缩短到负数的时间即可。

由于机会数不超过 k，时间复杂度为 $O\big((n + km)\log n \log(h_{\max} + ma_{\max})\big)$.