$Solution$

A. 铺设积木

见 NOIP2013/2018 Day1T1。

我们只讨论填平凹陷的部分。首先将高度进行差分，则目标是差分数组的所有数的绝对值均$\le k$。

每次选择一个区间 $+1$ 或 -1，等价于选择差分数组上的任意两个位置，将其中一个 $+1$ 另一个 -1。

于是我们可以计算所有超过 k 的减到 k 需要多少天，以及所有小于 $-k$ 的加到 $-k$ 需要多少天，取最大值即为答案。

时间复杂度 $O(n)$。

B. 开关灯

设 $n=p_1^{e_1}p_2^{e_2}\cdots p_n^{e_n}$，则 n 会在每一个 $\dfrac{n}{p_i}$ 时被翻转一次。

我们设 $f(n)=-1$ 表示开，$f(n)=1$ 表示关，则有 $f(n)=(-1)^n$。

最终答案即为 $\dfrac{n-\sum_{i=1}^n f(n)}{2}$，问题变为求 $f(n)$ 的前缀和。

由于 $f(n)$ 是一个积性函数，因此可以用线性筛在 $O(n)$ 的时间内求出，可以拿到 90 分。

剩下的 10 分是留给出题人的（笑），如果做出来了建议去备考 NOI。

C. 历史

首先，根节点的权值很好维护。接下来我们考虑从上到下经过一条边后点权的变化。如果能够维护所有边的权值，那么任意一个点的值都能通过点到根的和求出。

对于一次 1 操作，从 a 到根的路径上的边权值都会 $+1$，其余的边权值都会 -1。我们可以把 -1 看作 “背景”，在统计答案的时候手动减去 $depth\cdot num$，其中 num 表示至今为止 1 操作的数量。则问题转化为支持点到根加以及点到根求和，可以用树链剖分+树状数组维护。

另一个做法：对于任意一个点 x，它的权值为

$$\begin{aligned}\sum_a\left[v_a-len(a,x)\right]&=\sum_a v_a-\sum_a(dep(a)+dep(x)-dep(lca(a,x)))\\&=\sum_a v_a-\sum_a dep(a)-num\cdot dep(x)+\sum_a dep(lca(a,x))\end{aligned}$$

其中最后一部分可以通过树剖维护。

两种方法时间复杂度均为 $O(n\log^2 n)$。

D. 数字

数位dp。设 `f[i][j][k]` （其中 $0\le j,k\le 9$）表示当这个数形如 $a\times 10^i+k$，其中 a 的各位数字之和对 10 取模后为 j 的情况下，加到 $(a+1)\times 10^i+k'$ 需要多少步。相应地，设 `g[i][j][k]` 表示 k'。

`f[i]` 可以由 `f[i-1]` 迭代 10 次得到。

对于一个数 s，每一次我们在 n, m，这个数本身这三重限制下（下一次转移不能超过 n 步，数字不能超过 m，这个数形如 $a \times 10^i + k$），找到一个最大的 i，利用 `f` 和 `g` 进行转移。最后剩余一点的时候暴力转移。

这样，我们就可以在 polylog 时间内将一个数加到第一次超过 m。对每一个小于 10 的数都这么进行一次，记录一下它需要多少步超过 m，超过后对 m 取模余多少。两个结果分别记录在 `nxt` 和 `tot` 数组中。

然后，倍增处理 `nxt` 和 `tot`。设 `tot[d][i]` 表示从数字 d 开始，加到超过 m，反复 2^i 次需要进行多少步。相应地，`nxt[d][i]` 表示 2^i 轮后的数是多少。

计算答案分三步走：

- 找到最大的 i 将其进位，剩余一点的时候暴力转移，直到第一次超过 m；
- 从一个 < 10 的数开始，利用倍增处理加爆 m 的循环，直到最后一次加爆 m；
- 使用第一步的策略处理完 n 剩下的部分。