

A 奈乌部落

如果我们把怪物分成两组，进行生死擂台赛（即登场的怪物一直战斗到自己生命值为 0 或最后），那么最后剩下的怪物的生命之和为两组怪物血量之和的差。

对于每一种规定的任意对决模式，我们都可以画一张对决图，可以证明该图不存在环，于是可以证明任意对决图都是二分图，等价于分成两组怪物进行上述步骤。

所以我们使用背包 dp 计算某一组怪物的血量之和可以取到的值即可。

B 分蛋糕

答案即为 n 的 1 以外的因数个数

C 移动棋子

做法 1

贪心，能移动一定会移动。

考虑对于 1 号点的每个儿子为根的子树进行考虑，显然次数的瓶颈在于深度最深的点，那么先让一个深度最深的点一直移动直到进入 1 号点，之后每个时刻都会有点进入 1 号点了。

于是计算一下最深的深度 d ，子树内点的数量 c ，然后让某一个深度最深棋子畅通无阻得移动到 1 号点，计算这段时间内有棋子出去的时间戳数量，记为 a ，一棵子树的贡献就是 $d + c - a$ 。

然后对 1 号点的所有子树贡献取一个 max 即可，std 使用的是该思路。

做法 2

记深度 $\geq i$ 的棋子个数为 cnt_i ，深度 i 的贡献为 $cnt_i + i - 1$ ，对所有深度求最大值即可。

依旧是对 1 号点每个子树单独跑一边取最大值。

详细题解

独立的考虑每一个棋子，肯定是能走的时候一定会走，如果说这一步别的棋子走了导致自己不能走了，那就等一会走。容易想到贪心，这样一来答案就是每个棋子到 1 号点的距离加上等待的时间，在所有棋子里取最大值。考虑如何计算每个棋子需要等待的时间，这个过程比较困难，如果我们用全局的角度考虑，可以得出一些结论。

一个棋子肯定会在某个时刻进入一个连通块，连通块里的所有棋子会连续不断的进入 1 号点，并且连通块会在某些点进行合并，一个连通块的贡献为该连通块首次有棋子进入 1 号点的某个儿子的时刻，加上整个连通块的点的数量。这样的话可以考虑全局瓶颈，因为是能走就走，根据已经推出来的结论，如果我们钦定深度最深的棋子畅通无阻，那么全局的瓶颈肯定是深度最深的棋子所在的连通块。因为 1 号点的特殊性，我们取所有 1 号点的儿子为关键节点，答案是所有关键节点的贡献的最大值。而关键节点的贡献就是，每个节点子树内深度最深的棋子到该关键点的距离加上该棋子所带领的连通块点的数量。

由已推出的结论可得这样是正确的，那么这个连通块的点的数量怎么计算呢。首先深度最深的棋子是畅通无阻的，也就是说我们只有该点深度的时间去把别的棋子移到 1 号点使得它不会堵住深度最深的那个点。要考虑整个过程有那些点不属于深度最深的棋子的连通块貌似有点困难，但是时间戳和点是一一对应的，所以我们考虑哪些时间戳会有点出去就好了。于是 $O(n)$ 的复杂度通过此题。

D 小满的疑惑

第 n 天的任务只能第 n 天干，所以第 n 天一定干任务，之后发现，每往前推 2 天，就一定会有一天去完成任务，于是我们开一个小根堆倒着枚举，每次选当前堆里最轻松的一天完成任务。

也可以理解为，考虑现在后 $2k - 1$ 天有 k 天干任务，那么往前 2 天中有一天可以选择不干任务，另一天必须干任务，由于 $k = 1$ 满足，归纳法可以得证后续最优安排一定为上述过程。

E 数列变换

对原序列差分出来，把区间修改变成单点修改，即一个位置 $+1$ 同时另一个位置 -1 ，当原序列修改的区间端点为全局端点时，可以等价于差分序列某一个位置 $+1$ 或 -1 ，一字型对正负数分别求和取最大值，人字形枚举一个最高点，然后计算所有位置的贡献即可，最高点变换可以 $\mathcal{O}(1)$ 转移答案。

F 弹珠游戏

我们追踪球的路线，他一定是某一个方格的对角线，且不会存在某一个格子的两条对角线都经过（因为这种情况只会出现在墙壁在非整数坐标上），则小球在进洞之前的路径每一步都会经过不同的格子。我们考虑对于长边 n 小球会一直走到头再返回，并在另一维坐标上行动 $n \% m$ 或 $m - n \% m$ 个坐标，易知他会经历 $m / \gcd(n, m)$ 次长边的长度，答案为 $nm / \gcd(n, m)$ 。

G 放置哨兵

记 f_i 表示仅考虑前 i 个位置放置哨兵的方案数。转移的时候考虑枚举当前这个位置是否放置哨兵，容易发现如果当前位置不放置哨兵，会存在一个位置 j ， $[j, i]$ 这段区间均不放置哨兵非法，且这个 j 单调。

H 魔力串

由于子串一定是子序列，我们分别计算子序列和子串的数量相减即可，是两个很经典的 dp，当然也可以用 dp 实现，不过需要注意常数。

I 金银铜

如果存在两种等数量牌子一定可以达成目标（全部转化为另三种牌子）。

考虑有三种牌子，一次操作会使得某种牌子比其他两种牌子的差变化 3，于是我们发现如果存在两个牌子的数量差为 3 的倍数时，一定可以将他们俩的差调整至 0。

反之，我们记录每两种牌子数量差的集合 $\{d_{AB}, d_{BC}, d_{AC}\}$ ，他们只能进行 ± 3 操作，这相当于他们在对 3 取模意义下的值永远不变，故若在初始不存在 0 则永远无法运算出 0 来。