snake

考虑 $q=1$ ，先预处理出每个方格是否被毒蛇覆盖，若覆盖则设方格权值为 1 ，否则设为 0 。然后求解 $(1,1)$ 到 (n,n) 的最短路即可。用 dijkstra 复杂度则为 $\mathcal{O}(n^3+n^2\log n)$。

由于网格大小为 $n\times n$ ，那么 n 秒以后的网格都是一样的。所以 $q\neq 1$ 只需要对 $t=1,2\ldots n$ 分别求求解即可，时间复杂度 $\mathcal{O}(n^3+n^3\log n)$。

而方格权值要么为 0 ，要么为 1 ，使用 01BFS 即可使时间复杂度降为 $\mathcal{O}(n^3)$。

ride

对于 20% ，可以二分答案，设 $f(i,j)$ 表示当前位于点 i ，体力值剩 j ，路上经过的最大困难度。

对于 $m=n-1$ ，由于路径唯一，贪心选择最大边花费体力。

对于 $k=0$ ，求出最小生成树，答案就是 1 到 n 路径上最大值。

对于 100% ，二分答案 k，每条边的边权设为 $\max(0,v-k)$ ，然后跑一边最短路即可，时间复杂度 $\mathcal{O}(n\log^2 n)$。也可以使用桶优化去掉最短路的一个 $\log$。

perm

$f(i,j,S)$ 表示确定 $1,2,\cdots,i$ 在 b 中的位置，其中有 j 个 c 确定为 1（$i-j$ 个为 2），S 为剩余没填数的 b 位置集合。

1、$c=1 f(i,j,S)=\sum_{k\in S,|i+1-k|\le 1} f(i+1,j+1,S-\{k\})$

2、$c=2 f(i,j,S)=\sum_{k\in S,|i+1-k|\le 2} f(i+1,j,S-\{k\})$

注意到 $[i+3,n]$ 一定属于 S，并且 $[1,i]$ 的所有空位都是一样的，所以 S 只要记 $[i+1,i+2]$ 即可。复杂度 $O(n^2)$。

party

因为 2 操作是对整个连通块做贡献，所以使用并查集维护比较方便，记 cnt_i 为以 i 为代表节点的并查集举行了多少次聚会。

现在考虑 1 操作，操作相当于将两个并查集合并。假设将 Y 合并到 X 上，会发现此时 X 此前的聚会并未对 Y 进行贡献。要对 Y 的代表节点 y 再记一个 tmp_y，表示父亲节点在与自己合并之前，已经举行了多少次聚会，来起到一个差分的作用。

另外多出 1 点关系度的两人，可以用哈希表来记录。考虑查询，如果不在一个并查集里关系度肯定为 0。反之，则先暴力跳到他们的 lca，再暴力地往上跳，每次先加 cnt 再减 tmp。

最后的答案，还要先减去 lca 两个儿子 tmp 的较大值，再加上哈希表里二人额外的关系度。因为并查集树高是 log 级别的，所以暴力往上跳的复杂度是正确的。时间复杂度 $O(nlogn)$