

CodeM2018 初赛 B 轮题解

Light :

Idea :

先按记录的时间排个序, 然后按题意模拟工程师的进出过程, 同时维护每个房间的明暗状态。需要注意的是不能每次暴力赋值记录最后的答案, 因为这样做的时间复杂度是 $O(nmk)$ 的。我们要记录的是使得亮灯数最多的时刻, 最后复现一遍就好了, 时间复杂度 $O(nm + k)$ 。

Solution :

<https://paste.ubuntu.com/p/fNDqcVsZbr/>

Distribution :

Idea :

题目中时间的跨度只有 7 天, 因此在这段时间内, 所有公交线路加起来只会开 7m 次, 且我们能够知道哪些线路之间可以换乘。我们把这 7m 次行程按它们的时间排序, 用 $f[i][j]$ 表示完成前 i 个任务, 目前刚刚结束行程 j 时的最小花费, dp 即可。时间复杂度为 $O(nm^2)$ 。

Solution :

<https://paste.ubuntu.com/p/TxwVmtS2Xf/>

Lowbit :

Idea :

先对这个函数进行解释: 设 l, r 的前 k 位是相同的, 第 $k+1$ 位 l 是 0, r 是 1。那么 r 会把最低位的若干个 1 消除直到 $\log(\text{lowbit}(r))=k+1$ 后减 1。然后 l, r 的最长公共前缀增加, 重复这个过程。

为了让答案尽可能大, 显然 l 除掉前 k 位以后剩下的是 0000000....1(这样可以令最长公共前缀增加的最慢)。并且 k 肯定越小越好(这样该过程可以重复更多次)。从 r 的角度, r 除掉前 k 以后必然剩下 1111111....1。因此最优解取在 $l=1, r=111111111....1$ 上, 使用模拟或递推即可。

时间复杂度 $O(\log n)$ 或者 $O((\log n)^2)$

Solution :

<https://paste.ubuntu.com/p/zjF7pbsCrs/>

Plate :

Idea :

把一个点 (x, y) 翻到 $(-x, -y)$, 就可以把题意转换为在圆周上求一个点, 使得这个点到起点与终点的距离和最小。直接三分角度或二分导数就可以了。时间复杂度为 $O(nm)$ 。

Solution :

<https://paste.ubuntu.com/p/7n7twHBsq5/>

Sort :

Idea :

先假设 a 和 b 的差为奇数。这样我们其实可以交换任意差为奇数的两个石板。这是因为，我们总可以通过加魔法，将想要交换的两个数 c 和 d 的异或值($c \oplus d$)变为 255。这之后用异或魔法，就可以将 c 和 d 分别变为 0 和 255。最后逆向操作就可以将它们变为 a 和 b 并交换了。(交换后，再将 a 和 b 换回 c 和 d 。) 如果要交换差为偶数的两个石板，可以利用第三个和它们奇偶性不同的石板。这样我们就可以任意交换石板了，即可以做到任意排序。

若 a 和 b 的差为偶数，则我们永远无法交换写着奇数的石板 and 写着偶数的石板。那么，能完成整理的必要条件是，(a) 写着奇数的石板占据位置 1,3,5,7,...,255, 写着偶数的石板占据位置 0,2,4,6,...,254 ; 或者 (b) 奇数的石板占据位置 0,2,4,6,...,254, 写着偶数的石板必须占据位置 1,3,5,7,...,255。

类似的，若 a 和 b 的差的二进制末尾有 k 个 0 的话，我们永远无法交换模 2^k 不同的两个数。那么每个剩余类的石板必须占据某个剩余类（可能是自身也可能不是）的位置。

刚才已经证明如果 a 和 b 的差为奇数，所有的初始情况都是可以整理的。这就是说，每个剩余类内部都是可以排成升序的（忽略低位即可规约到差为奇数情况）。剩下的就是判断能否用加魔法和异或魔法将剩余类之间的位置调整正确。这里我们只考虑低位（如 $k=3$ ，只需考虑前 8 个数的最低 3 位）暴力搜索即可。如果只考虑低位无解，那么本组数据无解。如果只考虑低位有解，那么先让低位正确，再每个剩余类内部排序即可。

Solution :

<https://paste.ubuntu.com/p/bxbwM82gws/>

Sequence :

Idea :

考虑 dp 的做法，用 $f[i][k][l][x][y]$ 表示 b 的最后一位为 a 中的第 i 位， c 的最后一位为 a 中的第 j 位， b 的 size 的奇偶性为 k ， c 的 size 的奇偶性为 l ， b 违反规则 x 次， c 违反规则 y 次时的方案数。在不考虑任何优化的情况下，这是一个 $O(n^2)$ 的 dp，因为枚举到位置 p 时， $i=p-1$ 或 $j=p-1$ 成立，我们只要枚举另一个数字就行了。把 $i=p-1$ 和 $j=p-1$ 拆成两个数组看，并且把另一位看成是数字而不是位置，发现所有这些转移都可以通过线段树维护。

Solution :

<https://paste.ubuntu.com/p/vCmgSy4gGp/>