

Solution

第一次出题，真的不咋地。。。

题目真的是自己想的，如有雷同纯属悲剧。。。

P1. 新的世界

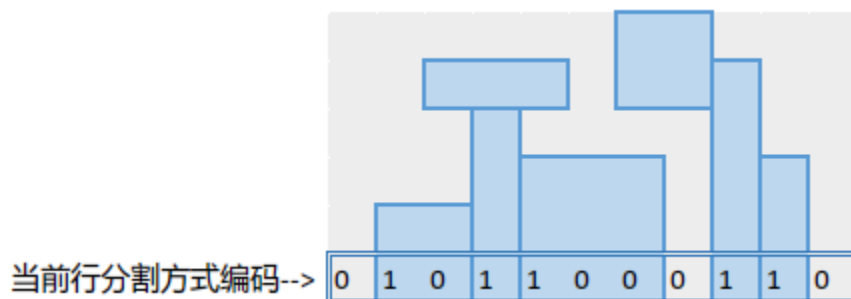
P1 题面居然是三题中最长的……（如果让你理解错了题意，我真诚地向你道歉 QAQ……）

想一想，这个“亮度更新”的过程是不是和 Bellman-Ford 的松弛操作有点像呢？不难发现， $\max\{\}$ 中的第 5 项只是保证最终亮度非负，而我们实际上可以暂时忽略它，先求出亮度衰减的最小值。如果对于一个方格 (i, j) ，让四周的方格向它连一条长度为 A_{ij} 的有向边，这个最小值正是 (r, c) 到 (p, q) 的最短路。由于 Bellman-Ford（也是题目描述中“游戏引擎”所使用的算法，其实改变计算的顺序不影响结果）和 SPFA 的时间复杂度不足以通过本题数据规模，而图中只有正权边，那就用 Dijkstra 吧！

PS：这是我第一次出题，自己 OI 水平也很差，没构造专门卡 SPFA 的数据……但还是推荐 Dijkstra 辣！

P2. 邻面合并

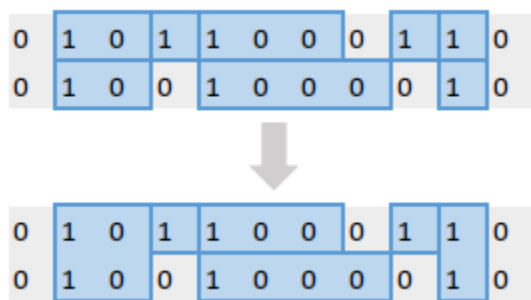
首先最暴力的搜索（每次枚举下一块长方形放在哪个区域）应该可以过 30% 的点吧？然后再看看 M 那么小……有没有想到状压 DP？所以考虑一行行转移，当前处理到第 i 行，最后一行的“分割方式”是 $mask$ 。如何表示？标记下每一块的开头就好辣！就像这样：



（数字表示编码，原矩阵的 0 和 1 仅用不同颜色表示）

（数据规模没这么大，图片仅供参考）

这样状态数就是 $O(N * 2^N)$ 了。转移的时候就枚举下一行的分割方式，然后计算为了达成这样的分割方式需要新创建多少个块。一个稍微简便的方法是用下一行的总块数，减去其中与上一行分割方式相同的块数（因为如果分割方式相同，两块可以合并），可以在 $O(N)$ 时间内完成……



（注意合并时需要同时考虑分割编码和原矩阵，比如这里的第 2 行第 2 块就无法与上一行第 3 块合并）

这就是 $O(N^2 * 2^{2N})$ 的做法…… $O(N * 2^N)$ 个状态， $O(N * 2^N)$ 的转移……

我也不知道有没有更快的做法……

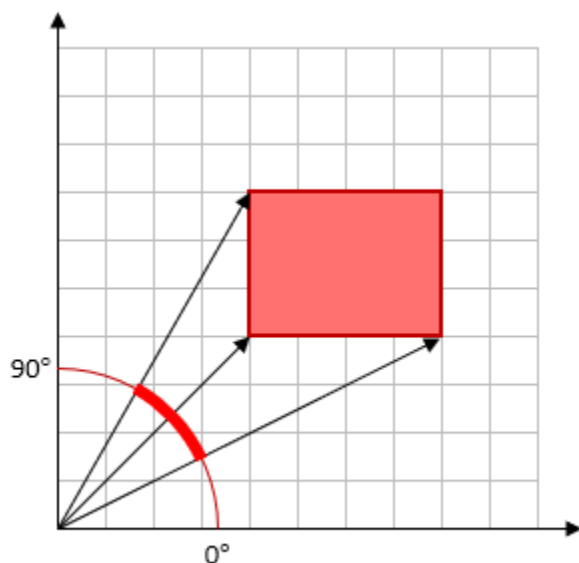
PS：这题题目描述是真是事，只是 NEWWorld 中遇到的数据规模是 $N, M \leq 32$ 的……（于是实际上我们采用了近似的启发式方法——贪心。。。不过很明显 OI 里不能这么做。。。）

P3. 光线追踪

首先注意，**长方形是闭区域**，射线碰到重边、交点的时候，要返回**最后覆盖上去（编号最大）**的长方形障碍的编号……比如样例 1 中的第 7 次查询，要输出 2 而不是 1。

前 30% 可以对所有障碍分别求交点，取最近的**那些障碍中编号最大的那个**；还有 30% 应该可以用“在边框上打标记”的做法处理询问 1，“一格一格向前走”处理询问 2……

如果我们再假想一下，从摄像机的视角“看”这个二维平面的第一象限会是怎样的？形式地说，有一个数轴，范围为 $0^\circ \sim 90^\circ$ ，其中位置为 a° 的点对应一个值，即动点前进方向与 x 轴夹角为 a° 时最早遇到的障碍物编号。这样每个长方形是不是就对应着轴上的一段区间了？



(这里将数轴画成弧形，加粗部分为对应区间)

可是如何维护长方形之间的互相“遮挡”关系呢……？由于从摄像机的视角“看”长方形，永远只能看到较近的两条边，那就只考虑这些线段；再把它们分成横边与竖边，这样一来，对于横边（或竖边），y（或 x）坐标更小的线段就肯定能在它所对应的数轴区间里遮挡住 y（或 x）更大的线段了。先把所有向量按与 x 轴的夹角排序，进行离散化，然后便是两个区间更新最小值、单点查询的问题了。用你喜欢的数据结构（比如线段树）维护一下就好辣！（注意 $x=0$ 或 $y=0$ 的情况……）

PS：这题坐标范围较大，可能会卡精度（虽然说数据没故意卡）……所以建议排序和二分查找（离散化）的时候用叉乘，而不用真的算出角度。同理，比较最近的横/竖边哪个更近时也可以保留分母然后十字相乘比较……标程中根本就没有用到浮点数哦！