

1. Our 题解

题目来源：未公开题目

大意：给一张有向二分图，一些点是不好的，现在从某个点出发，从左到右能够决策，从右往左随机一个点。问从哪些点出发可以一直只走好的点， $N, M \leq 10^6$

1.1:

首先那么如果左边某个点不好，那么右边能到这个点的点也可以视作不好，如果左边某个点到右边的所有点都不好，那么左边这个点也可以视作不好。于是可以每次暴力更新是否会有原来好的点被视作不好的点，直到没有新的点更新。根据实现获得 30/60 分。

1.2:

通过拓扑排序实现，记录每个点的入点，先将所有不好的点放入队列，每次拿出来，如果是左边的点就将右边的好的入点更新为不好的，然后放入队列。如果是右边的就更新左边的好的点的度数，如果左边好的点所有出点都不好，那么更新这个左边的点为不好。复杂度 $O(N + M)$ ，期望得分 100。

2. Mask 题解

题目来源：2024 ICPC昆明邀请赛 L

大意：二维平面上横竖有边，有一些斜边，求 $(0, 0)$ 到 $([0, p], [0, q])$ 间所有点的最短路之和

2.1:

暴力bfs,求出到每个点的距离，复杂度 $O(pq)$ ，期望得分30

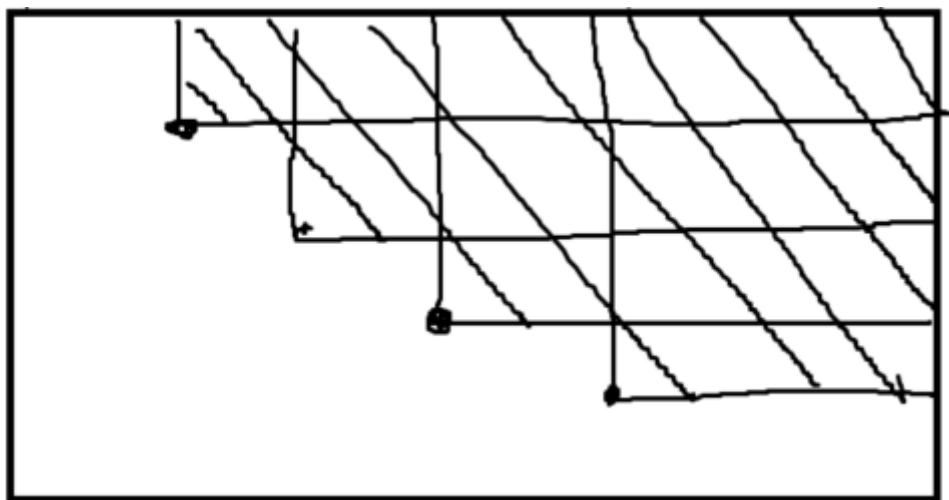
2.2:

考虑实际上答案是如何算出来的，如果没有斜边，那么 (x, y) 的最短路距离就是 $(x + y)$ ，只需要求向右上到 (x, y) 最多有多少条斜边就行

实际上只需要求 $\sum_{x=0}^p \sum_{y=0}^q (x + y) = (p + 1) * q * (q + 1) / 2 + (q + 1) * p * (p + 1) / 2$ 然后减去斜边数量。

如果斜边数量不多，可以考虑暴力对于每个斜边，求出走到这个斜边 i 前会走过多少斜边，设为 f_i 。对于每个格点，实际上也是查询这个格点左下方的最大的 f_i ，作为要减去的数量。

考虑所有相同的 f_i ，实际上这些斜边的分布一定是从左上到右下的一段折线。这些斜边产生的贡献就是斜边右上方的面积的并。



如果知道斜边，右上方的并的面积可以简单求出。暴力 $O(n^2)$ 求出每个斜边的 f_i 期望得分50.

2.3: 生成方式随机

可以发现复杂度瓶颈只在如何求出 f_i ，所以在随机数据下，可以尝试各种乱搞做法，比如每个斜边只向右上更新一部分，或者其他做法。

2.4:

考虑怎么简单求 f_i ，实际上就是一个二维数点，求左下部分的最大值，使用扫描线即可。用set或者树状数组，按 x 从左往右扫，维护竖着的前缀 max 即可。复杂度 $O(n \log n)$ 。期望得分100。

也可以直接在扫描线的过程中直接维护面积求和，具体实现可以参考代码。

3. Tale 题解

题目来源：2023 ICPC杭州站F，CF104976

大意：给一棵树，有互不相同的点权，多次询问 u, k ，询问离 u 距离不超过 k 的点点权集合的 mex ， $n \leq 5 * 10^5$

3.1:

暴力 dfs 处理每次询问，复杂度 $O(nq)$ ，期望得分20。

3.2:

n 仍然很小，对于每个点，将所有点按距离排序，预处理前缀 mex ，每次询问二分查询，复杂度 $O(n^2 + q \log n)$ ，期望得分 40。

3.3: 链

对于链的情况，问题变成询问区间 mex ，通过主席树等方式处理，复杂度 $O(n \log n)$ 。

3.4: 菊花

对于菊花的情况。分别考虑询问在叶子和 1 的情况，在 1 的时候，那么将所有叶子按照离 1 的距离排序，问题变成询问前缀 mex 。如果询问叶子节点，那么就相当与询问 1 和 $k - dist(1, u)$ ，额外考虑一个 u ，同样可以简单解决。

3.5: 做法一

考虑提示，那么我们只要求离 u 距离超过 k_i 的点里点权最小值，然后和没出现过的最小数值取 \min 。考虑点分树，先求出点分树，考虑每次询问。不断跳点分树上的祖先，考虑每个分治中心如何解决。即询问距离分治中心 x 距离超过 $k_i - dist(x, u_i)$ 的点权最小值，那么建点分树的时候把分治的这个子树所有点按照离分治重心的距离排序后记录后缀最小值，询问的时候二分查找即可。唯一一个需要额外考虑的问题是如何容斥掉 u_i 所在的子树的影响。一个办法是不光记录最小值，还记录一个和最小值不在一个同子树的次小值，那么答案一定是这两个值之一。再额外判断即可。

复杂度 $O(n \log^2 n)$ ，期望得分 80，如果实现优秀可能获得100。

3.6:

同样的考虑求离 u 距离超过 k_i 的点里点权最小值，考虑从最小值入手，将所有点按照点权从小到大排序，对于每个询问二分答案，需要判断点权不超过 mid 的所有点中是否存在一个点到 u 的距离大于 k_i 。

考虑预处理出每个前缀的直径的两端点，即前缀所有点里距离最远的两个点，然后维护前缀即加一个新的点，根据直径的性质只需要考虑新的点与原来直径端点之间三个点最远的距离即可。对于询问，显然只需要检查对应直径的两端点是否满足条件。使用 $O(1)LCA$ 来查询两点距离，总时间复杂度为 $O((n + q) \log n)$ 。

期望得分 100。

4. Box 题解

题目来源：2023JSCPC G

大意： n 个物品在一条直线上，每个需要从 x_i 移动到 y_i ，有一个机器人移动，机器人每次只能拿一个物品，速度为1，转向需要 c 秒，最后需要回到起点，方向回到初始方向。求运输完所有物品需要的最小时间， $n \leq 10^5$

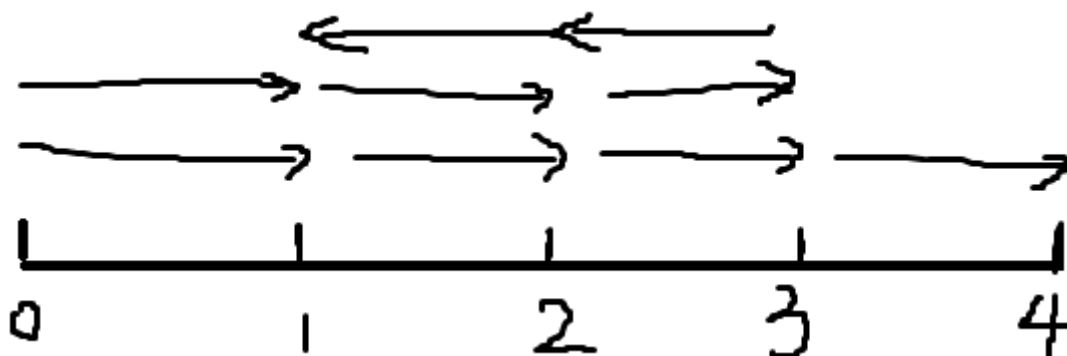
4.1:

首先可以爆搜，期望得分15。

4.2:

首先，由于物品可以中途放下，所以显然一条长路径可以看做若干单位长度的路径组成。先将所有的 x_i 到 y_i 拆分为若干单位长度的边。那么实际上就是要把这所有的边连接起来。例如， $x_1 = 0, y_1 = 4, x_2 = 0, y_2 = 3, x_3 = 3, y_3 = 1$ 就可以看做如下若干单位长度的边，我们要做的就是再加一些边

使得所有边能首尾相连成一个联通回路，代价是边长加上转弯的代价和。



实际上补上这些边即可使得所有边首尾相连。



由于要回到初始方位，我们一定可以从最小的 x_i 向右作为起点。

由于范围很大，离散化所有坐标。

我们考虑对于每个单位段，对于考虑有多少路径经过这个段，设从左往右经过第 i 个段路径个数为 a_i ，从右往左的个数为 b_i 。

当 C 很大, 坐标很小的时候, 显然我们会尽力避免转弯。所以一定是从最左边走到最右边, 每个单位长度的边能连就连, 直到再往前都没有边了, 就转弯回来, 继续。这时候耗时可以简单计算。期望得分30。

4.3:

考虑 $n \leq 2000$ 的时候可以怎么做, 实际上连接所有路径, 就是考虑在每个段的路径, 要么把向左和向右的, 用转弯连接起来, 要么向左的接上一个位置向左的, 向右的接下一个位置向右的, 可以直接链 dp , 设 $f[i][j]$ 表示前 i 段, 有 j 个路径还没连接, 考虑当前点, 是 x 向左, x 向右, y 向左, y 向右四种中的哪一个。然后考虑连接起来还是转弯, 分别讨论。具体实现可以参考 `std - 2k`。复杂度 $O(n^2)$, 期望得分60。

4.4:

当没有转向代价的时候, 显然随时想转就转。

设 $\phi(i)$ 表示最后答案之中从左往右第 i 段路径会被从左到右经过多少次, 由于会回到初始位置, 所以显然从右到左经过的次数是一样的。一个显然的结论是此时最优答案中, 每条边从左往右和从右往左都会经过 $\phi(i) = \max(a_i, b_i, 1)$ 次。暴力计算答案即可。可以获得额外的15分。

4.5:

设 $g(i) = \max(a_i, b_i, 1)$, 因为最后答案中每条边从左往右和从右往左一定会经过相同次数, 如果一条边没有被经过, 最后一定也会被经过至少一次 (否则, 这条边左右的边不连通), 所以我们可以先认为每条边都从左往右和从右往左都经过了 $g(i)$ 次, 显然加了这些边不会影响最后答案。

考虑第 i 和第 $i+1$ 和第 $i+2$ 个段路径之间, 设 $g(i) > g(i+1) < g(i+2)$, 如果转向代价为0, 那么这之间会转向 $g(i) - g(i+1) + g(i+2) - g(i+1)$ 次 (考虑从 i 到 $i+1$ 和 $i+2$ 到 $i+1$ 多余的那些路径, 会选择转向回去, 和反方向的路径接起来)。

如果转向代价无穷大, 对于 $i+1$ 这段, 会选择多走 $\min(g(i+2), g(i)) - g(i+1)$ 次, 使得 $\phi(i+1) = \min(g(i), g(i+2))$, 这样只需要转向 $|g(i) - g(i+2)|$ 次, 但是会多走 $\min(g(i+2), g(i)) - g(i+1)$ 次。

对于一般情况, 显然只需要比较两种方案的代价, 即比较 $\text{len}(i+1)$ 和 C 的值, 选择较小的方案即可, 这之后 $i+1$ 这个边就对左右不会产生任何影响了, 可以视作和 $i, i+2$ 里 g 较小的边合并了。

于是我们可以考虑一个过程, 从小到大枚举 g , 然后计算两种方案的贡献, 取较小的一个, 并和左右 g 较小的合并边长, 做到最后就是答案。

至于具体实现, 只需要对于所有的 $g(i)$ 做单调栈, 求出左右第一个 g 大于 i 的位置, 然后进行比较选择方案计算代价即可。

复杂度根据实现 $O(n \log n) / O(n)$, 期望得分 100。