

前言:

吴宫花草埋幽径，晋代衣冠成古丘。秋高气爽，桂花树下，又是一年挖坟时。

这套题由 lsq 爷爷搬运于 2016 年秋。

T1T2 的题解由本从来没有写过题解的蒟蒻负责，多多包涵 $\omega \cdot \text{)/}$ 。

---T1 能量异常---

题目大意：有 G (≤ 450000) 座城市，城市之间可能有边相连。询问从 1 号城市出发，到其余每个城市，最少需要经过几条边。

(1 号城市需要经过的边数视为 0；到达不了的城市需要经过的边数视为 -1)

首先需要发现的是，“两个城市同时发生异常”，核心是两个城市相关，连一条边。

共 20 个测试点。

算法一：对于测试点 1-3, $N \leq 10$ ，可以有各种奇葩的做法。期望得分 15 分。

算法二：对于测试点 1-8, $N \leq 1000$ 。

先对所有城市相关度赋值为 -1，byteland 为 0

重复进行以下操作：

扫描一次输入数据，寻找和相关度为 0 的城市有关联的城市，则可以确定这些城市的相关度为 1，不可能更小。

再次扫描所有数据，寻找和相关度为 1 的城市有关联的城市。如果这些城市当前相关度仍为 -1，则可以确定这些城市的相关度为 2，不可能更小。

... ..

重复进行该操作 3000 次（或者直到某次操作结束没有城市的相关度被更新），则可以确定所有城市的相关度，然后依次输出即可。

时间复杂度 $O(N^2)$ ，期望得分 40 分。

算法3：对于测试点 15，不难发现，1 号城市相关度为 0，和 1 号城市有关两个城市相关度为 1，其余城市相关度均为 -1。读入输入数据时记录，然后直接输出即可。

期望得分 5 分。结合算法二期望得分 45 分。

算法4：对于测试点 1-15, $N \leq 150000$

用邻接表存储边的关系。对于每一行输入数据，则每两个城市互相加入一条边（因为是双向的），一共加入六条边，不必考虑重边。

使用 vis 数组记录每个城市是否已被访问到，初始时 1 号城市为 1，其余城市为 0。

从一号城市开始 BFS，如果某个城市和正在处理的该城市有边，且没有被访问过，则加入访问队列。

（注意使用循环队列，或者将数组开得足够大；注意队列中的某个城市在开始时要判断是否被访问过，因为该城市在被加入时未被访问，但当处理到的时候可能已经访问过了；处理好的城市 vis 数组标位 1。）因为每条边会且只会被处理一次，时间复杂度 $O(N)$ 。

期望得分 75 分。

算法5: 对于测试点 16-20, $G_{i,j} \leq 10^9$

和算法 4 不同的是, 城市编号可能会很大, 因此要处理, 重定向编号后再进行加边, BFS。

~~好了好了知道你们是用 map 的了。~~

可以把所有读入的 $G_{i,j}$ 放在一个数组中, 对这些 (至多 450000 个) 数进行排序, 去重。即可建立数组 $city[n]$ 。其中 $city[i]$ 表示排序后编号 (从小到大) 的第 i 个城市对应的原编号。

随后进行加边时, 我们需要知道每个原来的城市编号所对应的现在的城市编号。使用二分即可。

(小优化: 空间换时间。对原来城市编号在一个范围内 (例如 ≤ 1000000) 的城市, 开一个数组, 直接对应他们现在的城市编号。这样可以不必每次都二分。)

接下来和算法 4 相同。记得输出时要输出原编号。题目要求按照原编号从小到大输出, 而排序后的新编号也是从小到大的, 正好满足。即: 原来编号最小的城市对应排序后的 1 号城市, 编号第二小的城市对应排序后的 2 号城市, 以此类推。

时间复杂度 $O(N \log N)$, 结合算法 4, 期望得分 100 分。

---T2 门泉探索---

题目大意:

给一个 $N \times M$ 的矩阵, 某些点有“友军的住所”, 某些点可以放 portals。Portal 可以覆盖周围曼哈顿距离不超过 R 的格子, 在覆盖范围内, 每有一个友军, 计数+1。同一个友军被多个 portals 覆盖可以重复被计数; 问如何使最后计数最大。

注意: 对边界需要进行处理; 或者选择不进行处理, 而把数组范围开的足够大。两种方法均可 AC, 但处理边界在数据较大时可以带来甚至十倍的速度提升。(代价是容易写错。)

算法一: 对于测试点 1-10, 暴力。

枚举 $N \times M$ 中 (可以放置 portal) 的每一个点, 再枚举该点周围曼哈顿距离不超过 R 的范围内有多少个友军, 得到 (如果在该点放置 portal) 可以获得的收益 $v[n \times 500 + m]$ (方便排序)。排序后选择从大到小的 T 个值, 相加即可。

时间复杂度 $O(N^4)$, 期望得分 50 分

算法二: 对于测试点 15-16, 暴力。

因为 $R=1$, 时间复杂度 $O(N^2)$, 期望得分 10 分。结合算法一, 期望得分 60 分。

算法三: 对于测试点 11-14, $N=1$

这是对标准解法的思维引导。我们发现, 在一行内, 可以先预处理“前 k 个格子里有多少个友军” (前缀和数组 $S[n]$)。随后对于 (可以放置 portal) 的每一个点, 在该点放置 portal 可以获得的收益不需要再枚举。若该 portal 的影响范围的最右边格子为第 R 个, 最左边格子为第 L 个。(注意边界) 可以直接用前缀和 ($S[R] - S[L-1]$) 算出该点的收益。最后收益排序相加, 同算法一。

当然, 暴力也能过。

时间复杂度 $O(N \log N)$ 或 $O(N^2)$, 期望得分 20 分。结合算法一和二, 期望得分 80 分。

算法四：对于测试点 1-20，二维前缀和。

在算法三的启发下，我们可以想到使用二维前缀和来优化寻找每个点若放置 portal 可以获得的收益。但是我们发现，一个 portal 的影响范围不是一个正的正方形，而是一个倾斜 45° 的正方形。实际上，这同样可以使用前缀和，只是条件更多，预处理（rong）更（yi）复（xie）杂（cuo）。

如图，我们想知道黄色正方形内有多少个友军，可以拿区块 1 减去区块 2 和 3，最后加上区块 4（这部分被区块 2 和 3 各减去了一次，因此需要加回）。 $S[i,j]$ 表示以 $[i,j]$ 为上顶点的斜 45° 正方形内有多少友军，思想和二维前缀和相同。注意边界，边界，边界。

时间复杂度 $O(N^2)$ ，期望得分 100 分。

