图论技巧与套路

Aftglw

2022 年 10 月 10 日

前言

题目难度与顺序完全无关!
题目顺序与难度完全无关!
题目难度完全无关于顺序!

Contents

SNOI2019 通信

5s, 512MB

有一个关键点 S 和排成一行的 n 个普通点，第 i 个普通点的权值为 a_i。

每个普通点有两种连边方式（有向边）:

1. 直接连向 S，代价为 W;
2. 连向普通点 $j(j < i)$，代价为 $|a_i - a_j|$。

每个普通点入度至多为 1。
要求从所有普通点出发都能到达 S，求最小代价。
$n \le 10^3, 0 \le W, a_i \le 10^9$。

SNOI2019 通信

5s, 512MB

考虑费用流。
将每个点拆为两个点 x_i 和 y_i。

- $s \to x_i$ 流 1 费 0，表示点 i 需要到达关键点；
- $x_i \to t$ 流 1 费 W，表示直接连向关键点；
- $x_i \to y_j$ 流 1 费 $|a_i - a_j|$，表示点 i 连向普通点 j，$(j < i)$；
- $y_i \to t$ 流 1 费 0，表示每个点入度至多为 1。

然后跑最小费用最大流即可，点数 $\mathcal{O}(n)$，边数 $\mathcal{O}(n^2)$，需要优化。
瓶颈在于第三类边。

SNOI2019 通信

5s, 512MB

由于有绝对值的存在，所以没法用线段树优化。

~~但其实可以通过分类讨论去掉绝对值，然后两棵值域线段树优化建图~~。

可以将这 n 个点复制一遍，然后按 a_i 排序，相邻两个点连流 ∞ 费 Δ 的双向边，构成一条链。

这样 i 连向 $[1, i-1]$ 的所有边都可以替换为 i 向链上连边，然后链向 $[1, i-1]$ 每个点连边。

但问题在于 i 也会被连，所以考虑 cdq 分治，每次复制一条长度为处理区间长度的链，然后通过上述方式让右区间向左区间连边。

这样点数边数均为 $\mathcal{O}(n \log n)$，可以通过。

NOI2019 弹跳

2s, 128MB

在一个 $w \times h$ 的网格上有 n 座城市，编号为 $1 \sim n$，每个城市的坐标均为整数，且互不相同。

有 m 个弹射装置，第 i 个弹射装置位于第 p_i 座城市，可以到达坐标满足 $L_i \le x \le R_i, D_i \le y \le U_i$ 的任意一座城市，耗时均为 $t_i(t_i > 0)$。

对于 $\forall i \in [2, n]$，求从第 1 座城市出发，到达第 i 座城市所需要的最短耗时。

$n \le 7 \cdot 10^4, m \le 1.5 \cdot 10^5, 1 \le w, h \le n, 0 \le t_i \le 10^4$。

注意空间限制。

NOI2019 弹跳

2s, 128MB

首先这题不难想到用 `KDTree` 、树套树或者二维线段树之类的优化建图，但是空间并不乐观。

考虑用一般的 `Dijkstra` 算法求最短路，我们是通过维护到某个点的最短路，再通过这个点的的出边去更新到其他点的最短路。

但这样我们发现一个点在更新其他点之前会被松弛很多次，因为一个点的入边会有很多。

但我们又发现一条边（一个弹射装置）只有一个起点，所以到一条边的最短路只会被松弛一次。

NOI2019 弹跳

2s, 128MB

如果我们用优先队列维护距离最短的边：

每次取出队首的边，首先这条边的终点的最短路已经确定了，因为这是距离最短的边。

由于终点的最短路确定了，所以终点的所有出边的最短路也确定了，可以入队更新其他边了。

那这样每个点只会被访问一次，每条边也只会入队出队一次，复杂度不就是 $\mathcal{O}((n+m)\log(n+m))$ 了？

还有个问题就是如何找到一条边所有的终点。

可以用树套树/KDT/二维线段树实现，每次找到一个点，松弛其所有出边后，就可以直接把这个点删除了。

所以总复杂度是 $\mathcal{O}(m\log^2 n+(n+m)\log(n+m))$ （树套树实现）。

CF1215F Radio Stations

7s, 256MB

有 p 个点，每个点有一个区间 $[l_i, r_i]$。
有 n 个形如 (x, y) 的限制，表示点 x 和点 y 中至少选一个；
有 m 个形如 (x, y) 的限制，表示点 x 和点 y 中至多选一个。
构造一种选择方案，满足上述的 $n + m$ 条限制，同时要求所有选择了的点的区间交集不为空。

$2 \le p, n, m, M \le 4 \cdot 10^5, 1 \le l_i \le r_i \le M$。

CF1215F Radio Stations

7s, 256MB

首先 $n+m$ 个限制很好做，可以直接 $2-\texttt{SAT}$。

要满足交集不为空，那么如果选了点 i，就不能选区间与 i 的区间不交的所有点。

如果将区间按照左端点排序，那么这些不能选的点就会是一段后缀 (虽然前面可能会还有一点但是不用管)。

于是可以对每个后缀建一个虚点，那么就变成了每个点和一个后缀的虚点至多选一个，还是 $2-\texttt{SAT}$。

点数边数均为 $\mathcal{O}(p)$，时间复杂度也是 $\mathcal{O}(p)$ 的。

Part I 小结

常见优化建图方式:

方式	点数	边数	优势（适用类型）
数据结构	取决于数据结构	取决于数据结构	区间向区间建相同的边，高效利用重复的边以减少边数
倍增	$\mathcal{O}(n\log n)$	$\mathcal{O}(n\log n)$	与线段树优化类似，但一般用于树上，且比线段树 + 树剖少一个 $\log$
前后缀	$\mathcal{O}(n)$	$\mathcal{O}(n)$	区间向区间外连边，常见于 $2-\mathtt{SAT}$; 也可配合树剖用于树上路径向路径连边，比线段树少一个 $\log$
cdq 分治	$\mathcal{O}(n\log^k n)$	$\mathcal{O}(n\log^k n)$	连边条件具有多重偏序关系
不建图	取决于题目	取决于题目	在空间限制较紧的情况下，直接让数据结构充当边的功能（查询边权，端点），以减小空间的开销

SCOI2007 修车

1s, 128MB

同一时刻有 n 位车主带着他们的爱车来到了汽车维修中心。

维修中心共有 m 位技术人员，第 i 位技术人员修第 j 辆车需要时间 $T_{i,j}$。

现在需要安排这 m 位技术人员所维修的车及其顺序，使得顾客平均等待的时间最小。

一位顾客的等待时间是指从他把车送至维修中心到他的车维修完毕所用的时间。

$2 \le m \le 9, 1 \le n \le 60, 1 \le T \le 10^3$。

SCOI2007 修车

1s, 128MB

“平均等待时间最小”等价于“总等待时间最小”。

对于一名技术人员 i 和他的维修序列 $a_1, a_2, \cdots, a_k$，贡献的总等待时间为：

$$\sum_{j=1}^{k}(k-j+1) \times T_{i,a_j}$$

那么将每个技术人员拆成 k 个点，表示其维修序列，每个点连向 t 流 1 费 0，表示一次只修一辆车；

第 i 辆车向第 j 个技术人员的第 k 个点连流 1 费 $(n-k+1) \times T_{j,i}$ 的边；

s 向每辆车连流 1 费 0 的边。

跑最小费用最大流即可，点数 $\mathcal{O}(nm)$，边数 $\mathcal{O}(n^2m)$，可以通过。

HNOI2007 紧急疏散 EVACUATE

1s, 128MB

给定一个 $n \times m$ 的房间，用矩阵 A 来表示。如果 $A_{i,j}$ 是 '.' 则表示一块空地；是 'X' 则表示一面墙；是 'D' 则表示一扇门，保证门只存在于边界上。

现在每个空地上都站着一个人，每个人都要从门出去，每个人每秒可以向四周非墙位置移动一格，或者不动。

一个空地上可以站无数个人，但一扇门每秒只能通过一个人。

你可以规划每个人的移动路线，求最小需要多少秒使得房间内没人，或者报告无解。

$3 \leq n \leq 20, 3 \leq m \leq 20$。

HNOI2007 紧急疏散 EVACUATE

1s, 128MB

二分答案 X。

由于每扇门每秒只能通过一个人，所以将每扇门拆成 X 个点，以表示每秒的门。

s 向每个空地连流 1 的边，每扇门的每个点向 t 连流 1 的边，表示每秒通过一个人。

还要将每扇门每秒的点向下一秒的点连流 ∞ 的边，表示等待。

可以先预处理出每个点到每扇门的最短路，就可以确定到达每扇门的时间，连向这扇门对应的点流 1 的边即可。

最后通过最大流是否等于总人数判断答案是否合法。

注意每次检验完之后删除新加的边。

复杂度 $\mathcal{O}($ 可过 $)$。

CF1368H1 Breadboard Capacity (easy version)

2s, 512MB

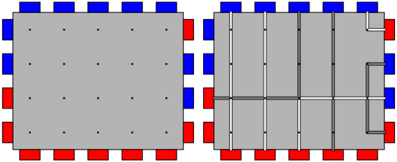
有一个 $n \times m$ 的网格图，边界上有红色或者蓝色的 port。

你需要架设管道，每个管道连接一个红 port 和蓝 port，管道必须水平或垂直，且两个管道之间不能有公共长度（可以有公共点）。

求最多能架设多少条管道。

$1 \leq n, m \leq 10^5$。

hard version：有 q 次修改操作，每次选择一条边界，将一段区间内的 port 颜色反转，计算每次修改后的答案。



CF1368H1 Breadboard Capacity (easy version)

2s, 512MB

首先可以想到 s 连向所有红色 port 流 1，所有蓝色 port 连向 t 流 1，然后沿着网格建边全部流 1，跑最大流即为答案。

然后发现过不去，考虑到最大流等于最小割，所以考虑最小割具体方案。

不难发现如果最小割是折线，那么可以不断调整来使得割边要么是一条长为 n 的横线，要么是一条长为 m 的竖线，要么在 port 与网格之间，这样调整后割边数量是不增的。

那么在最小割方案中，属于相同集合的点要么以一整行的形式呈现，要么以一整列的形式呈现，考虑 DP。

以一整行属于一个集合为例，设 $f_{i,0/1}$ 表示考虑前 i 行的划分，第 i 行属于 s/t 的集合，最小割边数量。

转移简单，同样对列再做一遍。

这样就可以通过 easy version 了，时间复杂度 $\mathcal{O}(n+m)$。

CF1368H2 Breadboard Capacity (hard version)

2s, 512MB

带上了修改操作，考虑 DDP。

发现转移可以用 2×2 的矩阵表示，于是用线段树维护。

考虑一整行属于一个集合的情况：线段树叶子节点维护 4 个 2×2 的矩阵，分别表示当前这行左端和右端的 port 都不反转颜色时的转移矩阵；左端反，右端不反转；左端不反、右端反转；左端和右端都反转，那么修改时就交换对应的两对矩阵即可。

一整列相同的情况类似。

还需要再用 4 棵线段树维护每条边界上的红色（蓝色）port 的个数，用来计算初始矩阵。

时间复杂度 $\mathcal{O}(q \log n)$

CF802C Heidi and Library (hard)

2s, 256MB

你有一个容量为 k，初始为空的书架，书店有 n 种书，第 i 种书的价格为 c_i。

接下来 n 天里，每天会给定一个 a_i ，要求这天书架上必须有第 a_i 种书。

任何时刻都可以从书架上扔掉一本书，也可以从书店买一本书放在书架上，但不能将扔掉的书放回书架上。

在满足所有要求的情况下，求最小花费。

$n, k \leq 80, c_i \leq 10^6$。

CF802C Heidi and Library (hard)

2s, 256MB

先构造一个可以满足所有要求的方案：第 i 天买第 a_i 种书，用完就扔掉。

那么现在的问题就成了尽可能地减小花费。

考虑两天 i 和 j，如果这两天的要求是一样的，也即 $a_i = a_j$，那么就可以在第 i 天用书完之后不扔，然后第 j 天就不用买了。

这样可以减小 c_{a_i} 的花费，但是会在第 i 天到第 j 天占用书架 1 的空间。

那么问题就变成了：有一个长度为 n 的数轴和一些区间，每个区间有一个权值，选取若干区间并在数轴上进行覆盖并获得权值。在数轴上每个位置被覆盖不超过 k 次的前提下，求能获得的最大权值和。

这是一个经典的区间 k 覆盖问题，可以通过费用流解决。

点数和边数均为 $\mathcal{O}(n)$ 的。

CF802C Heidi and Library (hard)

2s, 256MB

区间 k 覆盖问题的费用流解法：
先假设所有区间都选上。
新加一个点 $n+1$，这样就可以将点的覆盖限制转化到边的流量限制上。

- 第 i 个点向第 $i+1$ 个点连流 k 费 0 的边，表示点的覆盖限制；
- 对于一段区间 $[l_i, r_i]$ 权值为 w_i：
  - s 向 l_i 连流 1 费 0 的边；
  - r_i+1 向 t 连流 1 费 0 的边。
  - l_i 向 r_i+1 连流 1 费 $-w_i$ 的边；

那么一段区间的流量，要么走 $l_i \to l_i+1 \to \cdots \to r_i+1$ 费用为 0，但占用了包含的每个点 1 的流量；要么走 $l_i \to r_i+1$ 费用 $-w_i$ ，但不占用流量。
跑最大费用最大流即可。

Part II 小结

拆点（加点）是网络流建模中必不可少的一个环节，一般的动机来自于：1. 将对点的限制/费用转化到边上（这个很常见）；2. 表示更多的状态（比如 SCOI2007 修车中由于修车的顺序不同会导致费用的不同，所以将“一个人”拆成“修第几辆车的人”；HNOI2007 紧急疏散中由于每扇门每秒只能通过一个人，但一个点一条边只能限制“每扇门”和“一个人”，所以拆成“每秒”的门来限制时间）。

网络流中部分题目会与二分答案相结合：有的会用最大流根据最大流是否为总流量来判断（HNOI2007 紧急疏散）；有的会用费用流根据费用是否非负或者是否发生变化来判断（SDOI2017 新生舞会）；有的会用上下界网络流根据是否存在可行流来判断（LuoguP4194 矩阵）。这类题目出去一般会给大量的时间用于判断答案，因为网络流本身复杂度较高，且每次判断都需要重构图。

Part II 小结

有些题目一看就是很显然的最大流/费用流，但是一看数据范围就是很显然的做不了。这类题目可以考虑模拟网络流过程，比较常见的是模拟费用流，得到反悔贪心的做法；当然还有模拟最大流的，将考虑手动划分最小割（CF1368H Breadboard Capacity）。

至于区间 k 覆盖类问题，属于是网络流中的一个套路，还是可以积累一下，说不定哪天就用上了。

SCOI2011 糖果

1s, 128MB

有 n 个数，分别为 $a_{1\sim n}$，有 m 条限制，每条限制形如 (X_i, A_i, B_i)。

- 若 $X_i = 1$，则要求 $a_{A_i} = a_{B_i}$；
- 若 $X_i = 2$，则要求 $a_{A_i} < a_{B_i}$；
- 若 $X_i = 3$，则要求 $a_{A_i} \geq a_{B_i}$；
- 若 $X_i = 4$，则要求 $a_{A_i} > a_{B_i}$；
- 若 $X_i = 5$，则要求 $a_{A_i} \leq a_{B_i}$。

构造一种 a，使得 $a_i > 0$ 且 $\sum a_i$ 最小，或者报告无解。

$n, m \leq 10^5$。

SCOI2011 糖果

1s, 128MB

这题 SPFA 被卡掉了。

若要求 $a_i = a_j$ ，则可看作 $a_i \leq a_j \wedge a_i \geq a_j$。

若要求 $a_i \geq a_j$ 或者 $a_i > a_j$，则连边 $i \to j$。

建出的图后，考虑缩点。

每一个强连通分量中，如果都是 $\geq$ 边，那么这个强连通分量内所有点取值都相等；否则就无解。

缩点后得到 DAG，那么按拓扑序倒序依次贪心地给每个强连通分量赋值即可。

时间复杂度 $\mathcal{O}(n+m)$

CF555E Case of Computer Network

3s, 256MB

给定一个 n 个点 m 条边的无向图和 q 个有向点对 (s_i, t_i)。
试图给每条边定向，使得对于每个有向点对都满足：s_i 能到达 t_i。
判断是否有解。
$n, m, q \leq 2 \cdot 10^5$。

CF555E Case of Computer Network

3s, 256MB

首先有个结论：在无向图的每个边双中，可以通过给边定向使得分量中任意两点间互相可达。

一种理解方式是：边双中每条边都在于环中；

另一种理解方式是：可以按以下方式给边定向，使得边双转化为强连通分量：

- 先整出 `dfs` 树（外向），然后对于每条返祖边都定为指向祖先的。

可以证明这样得到的图是强连通的。

所以边双缩点，然后原问题就转化为了给树定向。

由于树上路径是唯一的，所以每个点对相当于限制一条路径上边的方向，可以树上差分解决。

时间复杂度取决于求 `LCA` 的方式。

GYM102059B Dev, Please Add This!

2s, 256MB

有一个 $n \times m$ 的网格，每个格子要么是障碍，要么是空地，要么是有奖励的空地，其中有一个空地上有一个球。

你可以控制球向四方移动，但一旦选定一个移动方向，那么球会一直沿着这个方向移动直到碰到障碍或边界。

询问是否可以吃到所有奖励。

$n, m \leq 50$。

GYM102059B Dev, Please Add This!

2s, 256MB

首先发现，对于横着/竖着的一段联通区域，可以操纵小球反复横跳全部经过。

那么就可以对每一段横着/竖着的建一个点，然后发现每一个横点可以到达两个竖点，每一个竖点也可以到达两个横点，然后就可以得到一张有向图。

那么我们做的就是要从起点的横点或竖点出发，在这张有向图上走。

考虑每个点是否到达，用 true/false 表示。

1. 对于奖励格，则要求其对应的横点和竖点至少经过一个，有点 $2-\mathtt{SAT}$ 的感觉了，于是 $\mathtt{false(a)} \rightarrow \mathtt{true(b)}, \mathtt{false(b)} \rightarrow \mathtt{true(a)}$
2. 对于从起点无法到达的点，那么它只能是 false ，于是 $\mathtt{true(a)} \rightarrow \mathtt{false(a)}$
3. 对于两个点 (a, b) 如果 a 不能到达 b ，且 b 不能到达 a ，那么 a, b 不能同时经过，$\mathtt{true(a)} \rightarrow \mathtt{false(b)}, \mathtt{true(b)} \rightarrow \mathtt{false(a)}$

GYM102059B Dev, Please Add This!

2s, 256MB

然后跑一遍 $2-\texttt{SAT}$ 来判断是否有解。

首先充分性显然，这个 $2-\texttt{SAT}$ 无解那么原题无解。

考虑证明必要性，也就是构造原题的一组路径来经过 $2-\texttt{SAT}$ 的解中每个为 true 的点。

根据第 3 类边，我们发现对于任意两个 u, v ，如果都取了 true ，那么也就表明要么 u 一定存在一条到 v 的路径，要么 v 就一定存在一条到达 u 的路径。这是一个很强的性质，不难发现这对应着一张竞赛图。

所以原题所求就是一条竞赛图上的哈密顿路径，这是必然存在的。

所以 $2-\texttt{SAT}$ 有解等价于原题有解。

时间复杂度 $\mathcal{O}(nm)$。

AGC036D Negative Cycle

2s, 1GB

有一个 n 个点的带权有向图，有以下三类边：

1. $i \to i+1$ 边权为 0;
2. $i \to j, (i < j)$ 边权为 -1;
3. $i \to j, (i > j)$ 边权为 1。

给定一个矩阵 A，其中 $A_{i,j}$ 表示删去有向边 $i \to j$ 的代价。
你需要删去若干条 $2, 3$ 类边使得图中不存在负环。
求最小代价。
$3 \leq n \leq 500, 1 \leq A_{i,j} \leq 10^9$。

AGC036D Negative Cycle

2s, 1GB

差分约束无解 $\Leftrightarrow$ 存在负环；不存在负环 $\Leftrightarrow$ 差分约束有解。

对于 i 连向 $i+1$ 的边权为 0 的边，相当于满足不等式 $x_i+0 \geq x_{i+1}$ ，也即 $x_i - x_{i+1} \geq 0$。

设 $p_i = x_i - x_{i+1}$，由于 0 边是不能删的，所以必然有 $p_i \geq 0$。

考虑保留一条 -1 边 $i \to j(i<j)$ ，那么需满足 $x_i - 1 \geq x_j$ ，也即 $p_i + p_{i+1} + \cdots + p_{j-1} \geq 1$。

那么 $\sum p = 0$ 意味着这条 -1 边被删除，也就是说 p 中的一个全 0 的段对应着若干条 -1 边的删除。

同理，考虑保留一条 1 边 $i \to j(i>j)$ ，那么需满足 $x_i + 1 \geq x_j$ ，也即 $p_j + p_{j+1} + \cdots + p_{i-1} \leq 1$。

那么 $\sum p \geq 2$ 意味着这条 1 边被删除。为了让代价尽可能小，我们要尽量少删边。而如果 $p_i \geq 2$ 那么就意味着边 $i+1 \to i$ 会被删除。这显然是不优的，因为我们可以让 $p_i = 1$ 来保留这条边，而不删去其他任何一条边。

AGC036D Negative Cycle

2s, 1GB

那么现在问题转化为构造一个长度为 $n-1$ 的布尔序列 p ，每段 0 都会产生一些代价，每一段区间和 ≥ 2 的区间也会产生代价，求最小代价。

这个问题可以考虑 DP。

设 $f_{i,j}$ 表示前 i 个位置，第 i 个位置是 1，且上一个 1 在 j 时的最小代价。

$f_{i,j}$ 由 $f_{j,k}$ 转移过来，新增的代价为左右端点在 $[j+1,i]$ 的 -1 边和左端点在 $[k+1,j]$ ，右端点在 $[i+1,n]$ 的 1 边，可以通过预处理前缀和实现 $\mathcal{O}(1)$ 转移。

时间复杂度 $\mathcal{O}(n^3)$。

JOISC 2020 Day4 治疗计划

2s, 512MB

有 n 个房屋排成一行，每个房屋都有一个新冠患者。

每天早上，如果某个房屋有新冠，那么到了中午，与它相邻的没有新冠的房屋就会被传染。

有 m 个治疗方案，第 i 个治疗方案代价为 c_i，会在第 t_i 天晚上治好 $[l_i, r_i]$ 范围内的房屋里的所有人。

求最小代价使得所有人都没有新冠，或判断无解。

$1 \le n, t_i, c_i \le 10^9, 1 \le m \le 10^5$。

JOISC 2020 Day4 治疗计划

2s, 512MB

考虑 DP，设 f_i 表示选取若干方案，治好前 r_i 个人的最小代价。

尝试向右扩展至 $r_j(r_j > r_i)$，有转移 $f_j \leftarrow f_i + c_j$，限制条件是 $l_j \leq r_i - |t_i - t_j| + 1$，也就是两个治疗方案要衔接上，不让中间出现病例。

注意到，每个方案至多被选取一次，并且转移的代价是与终点相关的（也就是说 f_j 的最小值一定是由一个最小的 f_i 转移来的）。

所以可以用类似 `Dijkstra` 的方式进行贪心转移。

至于限制条件，可以分类讨论去掉绝对值，然后然后用线段树维护区间最小值。

由于每个方案入队出队一次，再根据势能分析，总时间复杂度 $\mathcal{O}(m \log m)$。

Part III 小结

当差分约束的边权变为 $0, 1, -1$ 时，相比寄掉了的 SPFA，我们可以选择更优秀且稳定的 Tarjan + 拓扑序上 DP 来代替（SCOI2011 糖果，POI2012 FES-Festival）；

无向图给边定向，一个结论是：一个边双可以通过给边定向满足这个连通块在有向图状态下是强连通分量。于是可以按边双缩点，然后转化为树上问题。（CF555E Case of Computer Network，CF732F Tourist Reform）

负/正环对应着差分约束的有解性，于是可以反向利用差分约束来解决负/正环相关的问题。（AGC036D Negative Cycle）

对于 DP 只要满足：转移的代价只与转移的终点相关，那么就可以通过 `Dijkstra` 的贪心思想来优化 DP。（JOISC 2020 Day4 治疗计划）