图论

___debug

2018 年 7 月 7 日

Broken

给你一个 N 个点的无向完全图, 每条边有边权.

要你把这张图分成两部分, 使得每部分内部的边权最大值加起来最小.

$$N \le 300, 0 \le w_i \le 10^9$$

考虑枚举两部分的边权最大值 a 和 b, 那么如果一条边 (u, v) 的权值大于 a, 那么这两个点就不能同时出现在 A 集合中; B 集合同理.

那么我们可以将这个转化为一个 2-SAT 问题, 并且这个 2-SAT 问题只需要判定是否有解, 所以直接强连通分量即可.

显然 A 集合的最大边枚举, B 集合的最大边二分, 我们可以做到 $O(N^4 \log N)$. 那么现在我们只能优化 A 集合的边枚举.

考虑从大到小枚举, 同时维护图的连通性. 那么如果一条边加入后形成了一个偶环, 那么这条边一定不会成为 A 的最大边.(因为如果这条边被包含进 A 了, 那么一定至少有另一条边也被包含了) 如果形成的是奇环, 那么这条边一定是 A 的最大边的下限 (即不可能比这个更小了).

所以我们对于一条树边, 直接做; 对于一条形成偶环的边, 不管; 对于一个形成奇环的边, 做完以后直接退出即可.

复杂度 $O(N^3 logN)$.

tournament

一个 N 个点 $N(N-1)/2$ 条有向边的竞赛图 (每对点之间都有连边), 现在随机给每条边定向, 问 1 号点所在的强连通分量的期望大小.

$N \le 1000$

考虑竞赛图的一个性质: 对一个竞赛图拓扑排序, 每一层的点一起组成了一个强连通分量.

所以这道题递推枚举的思路为考虑拓扑排序中最后一层点, 也就是最后一个强连通分量. 由于每个点最后所在强连通分量的期望大小是一样的, 不妨求出每种方案中每个点所在强连通分量大小的和的和, 最后除以点数和方案数之积即可.

设 $g(n)$ 为 n 个点时, 每种情况中每个点所在强连通分量大小的和的和, 则有

$$g(n) = \sum_{i=1}^{n} \binom{n}{i} f(i) \left(g(n-i) + 2^{\frac{(n-i)(n-i-1)}{2}} i^2 \right)$$

(枚举最后一个强连通分量的大小为 i)

其中 $f(n)$ 为 n 个点的竞赛图中, 整张图是强连通图的个数.

$$g(n)=\sum_{i=1}^{n}\binom{n}{i}f(i)\left(g(n-i)+2^{\frac{(n-i)(n-i-1)}{2}}i^2\right)$$

其中 $f(n)$ 为 n 个点的竞赛图中, 整张图是强连通图的个数. 显然有

$$2^{\frac{n(n-1)}{2}}=\sum_{i=1}^{n}2^{\frac{(n-i)(n-i-1)}{2}}\binom{n}{i}f(i)$$

(枚举最后一个强连通分量的大小为 i)

然后就有递推式

$$f(n)=2^{\frac{n(n-1)}{2}}-\sum_{i=1}^{n-1}2^{\frac{(n-i)(n-i-1)}{2}}\binom{n}{i}f(i)$$

$O(N^2)$ 递推解决.

city

给你一个 N 个点 M 条边的无向图, 每条边有边权. 现在有 Q 个询问, 假设现在图中一条边也没有, 每次询问要使 u, v 在同一个边双连通分量中至少需要加入的边的权值的下界, 强制在线.

$N, Q \leq 10^5, M \leq 10^6$

先求出最小生成树, 如果 u 到 v 的路径上没有桥就一定有解.
对于一条非树边 (u,v), 将 u 到 v 的路径取一个 $\min$ 就可以了.
LCT 或者树链剖分均可.

[Codeforces 97E] Leaders

给你一个 N 个点 M 条边的无向图，无重边无自环，Q 次询问，每次询问点 u, v 之间是否存在一条长度为奇数的简单路径.

定义简单路径为不经过重复的点的路径.

$N, M \leq 10^5$

首先求出一个生成树, 不妨假设整张图是连通的.

引理 1:

两个在生成树上的距离为偶数的点 u, v 之间存在至少一条长度为奇数的简单路径, 当且仅当这两个点在生成树上的路径上存在至少一条边在某个奇环上.

引理 2:

一个点双连通分量中要么每条边都在至少一个奇环上, 要么没有奇环.

判断一个点双连通分量中是否存在奇环, 只需考虑是否存在一条非树边连接了两个深度奇偶性相同的点即可.

travel

给你一个 N 个点 M 条边的连通无向图, 每个点有点权. Q 次操作, 分两种:

1. 询问所有 u 到 v 的简单路径 (即不经过相同的点) 中, 所有可能经过的的点中点权的最小值
2. 修改某个点的点权

$$N, M, Q \leq 10^5, w_i \leq 10^9$$

首先计算点双连通分量.

由于割点会出现在多个点双中, 所以不妨将其独立出来连成一棵树.

修改的时候会出一些问题. 比如说我们用 multiset 维护每个点双中点的权值. 修改割点权值的时候, 如果把每个包括它的点双都改一遍, 这样复杂度就有问题了. 可以这样处理: 即如果改的是割点, 只维护它父亲的 multiset. 查询的时候, 如果两个点的 LCA 的父亲是割点, 那么就将其也考虑进去.

sleet

有 $2n$ 个人, 要安排座位围成一圈. 编号为 x 的顺时针方向下一个人编号必须是 $2x, 2x+1, 2x-2n, 2x+1-2n$ 中的一个. 输出任意一种方案.

$n \le 5 \times 10^5$

考虑 $x(x \leq n)$ 和 $x+n$ 这两个点, 它们对下一个人的要求是相同的. 所以我们把它们用一个点表示, 然后用边表示这些限制.

做一遍欧拉回路即可.

reality

给你一个 n 个点 m 条边的有向图，问单独删去哪些点之后会变成一个 DAG.

即求哪些点在图中所有环的交中.

$n \le 5 \times 10^5, m \le 10^6$

不妨先特判掉无解的情况. 然后可以随便找到一个环, 我们要找的点一定在这个环上, 并且删去这个环后原图变为 DAG.

不妨把每个点拆成两个点 u_1, u_2, u_1 只连进来, u_2 只连出去, 然后 $u_1 \to u_2$. 问题转化为删去哪些环上的边后图一定变为一个 DAG.

首先从某个点开始, 对环上的点编一个号, 使得 $1 \to 2 \to \cdots p \to 1$. (断环为链) 然后讨论两种路径:

1. 从 id 小的点连向 id 大的点的路径
2. 从 id 大的点连向 id 小的点的路径

显然第 1 种路径会导致中间一段的点都不合法, 而第 2 种则会导致除了中间一段的点都不合法.

我们可以在拓扑序上 DP 求出 $mx(u)$, 表示 u 点通过一些边能到达的环上 id 最大的的点. 然而我们只能利用 $mx(u)$ 求出第 1 中路径的影响, 而不能求出第 2 种的. 因为对于一个点 u, 我们必须知道在它能到的节点中, $\mathrm{id}(v)$ 最大且满足 $\mathrm{id}(v) < \mathrm{id}(u)$ 的 v, 这样才能知道情况 2 的最大的不合法区间. 这样并不好做.

不妨这样考虑: 只要一个环上的点可以走到 id 比它小的点, 那么 id 比它大的点一定不合法了; 同理, 只要一个环上的点可以被 id 比它大的点走到, 那么 id 比它小的点一定不合法了.

所以我们只需再求出 $mn(u)$ 表示 u 点通过一些边能到达的环上 id 最小的的点, $rmx(u)$ 表示 u 点逆着一些边能到达的环上 id 最大的点, 即可解决情况 2.

时间复杂度 $O(n + m)$.

给你一个 n 个点 m 条边的无向图, 统计四元环的个数.

$n \leq 50000, m \leq 10^5$

性质:

在一个 m 条边的图中, 与任意一个点相邻的且度数不小于它的点的个数至多为 $\sqrt{m}$.

定义每个点的优先级为它的度数, 度数大的点优先级高, 如果两个点度数相同则比较编号大小.

枚举 u, 再枚举 (u,v). 依次对于所有满足 w 的优先级大于 u 和 v 的边 (v,w) 统计答案. 这样, 我们只会在每个环的优先级最大的那个点把它算到.

如果我们与先处理出一个图 G' 只包含满足 u 的优先级小于 v 的边 (u,v), 时间复杂度可以做到 $O(m^{3/2})$.

[Codeforces 701F] Break Up

给你一个无向图，边有权值．要你删去最多两条边，使得 S 到 T 不连通，且删去的边的边权和最小．

$N \leq 10^3, M \leq 3 \times 10^4, 1 \leq w_i \leq 10^9$

先以 S 为根求出生成树. 首先考虑删去一条边的情况, 此时删去的边一定是 T 到 S 的那些树边, 并且一定是桥.

然后对于删去两条边的情况, 其中一条也一定是上面所说的树边. 不妨枚举这条边, 然后再对新图做 “删去一条边” 的判断即可.

时间复杂度 $O(NM)$.

[BZOJ 2654] tree

给你一个无向带权连通图，每条边是黑色或白色．求一棵最小权的恰好有 t 条白色边的生成树．

$N \leq 5 \times 10^4, M \leq 10^5$

二分出最小的 x, 满足将每条白边减 x 后求出的最小生成树 (相同权值选白色边) 白边个数 $\geq t$.

如果每条白边减去 x 后算出的最小生成树恰好选了 t 条白边, 那么这个生成树就是答案.

如果每条白边减去 x 算出的最小生成树中的白边数 $> t$, 而减去 $x-1$ 时的最小生成树中白边数 $< t$, 减去 x 时算出的那个最小生成树就是答案.

所以最后答案就是 $W - xt$, 其中 W 为最小生成树的权值.

[BZOJ 2521] 最小生成树

对于某一条无向图中的指定边 (a, b), 求出至少需要多少次操作可以保证 (a, b) 边在这个无向图的最小生成树中.

一次操作指: 先选择一条图中的边 (u, v), 再把图中除了这条边以外的边, 每一条的权值都减少 1.

$$N \le 500, M \le 800, 1 \le w_i < 10^6$$

首先所有其他边权值减小 1 就相当于自己的权值增加 1.

而我们要求所有 a 到 b 的路径上的最大边大于 (a,b) 的权值. 不妨给每条边一个新边权

$$w_i' = \max\{w(a,b) - w_i + 1, 0\}$$

然后做一遍最小割就即可.

save

给你一个 N 个点 M 条边的带权无向图；同时给你一个参数 T
.

一共 Q 个询问，一个询问三个参数 d, u, s ，询问在时刻 d 时，从 u 出发经过边权不超过 s 的边组成的连通块，这个连通块里未被覆盖的点的个数；同时，这个询问会将这个连通块里的点全部覆盖.

一个点若最后一次被覆盖时间是 t ，则其会在 $t+T$ 时刻变回未覆盖状态. 询问按 d 从小到大输入.

$$N, Q \leq 10^5, M \leq 2 \times 10^5, T \leq 10^8$$

考虑把边和询问按权值大小排序. 如果 $T=1$, 即每次覆盖就立即消失的话, 就可以直接扫过去用并查集做.

对于原问题, 考虑把一条边也看作一个点, 那么合并两个连通块实际就相当于把这两个连通块都连向一个点. 这样, 如果把最后的连出来的树做一遍 DFS, 那么每次询问的点就代表了一段连续的区间.

接下来就很好做了. 对于每个询问有两个事件, 用一个线段树维护每个点当前被多少个询问覆盖, 同时支持查询有多少个 0 就可以了.

[BZOJ 2069] [POI 2004] ZAW

给定一个有 N 个点 M 条边的无向图, 每条无向边最多只能经过一次. 对于边 (u,v), 从 u 到 v 的代价为 a, 从 v 到 u 的代价为 b, 其中 a 和 b 不一定相等.

求一个包含 1 号点的有向环, 使得环上代价之和最小.

$N \le 3 \times 10^4, M \le 10^5, 1 \le a, b \le 10^4$, 保证没有重边和自环

考虑一条包含 1 的有向环, 一定是 $1 \to x \to \cdots \to y \to 1$ 这样子.

我们当然可以枚举 x, y 然后做最短路, 但是这样显然太慢了.

但是这里的最短路是可以 “并行” 地求的. 也就是说, 如果给定两个不相交的点集 $\mathcal{A}, \mathcal{B}$, 那么我们可以用一次最短路的时间求出所有点对 (x, y) 满足 $x \in \mathcal{A}, y \in \mathcal{B}$ 的最短路的最小值.

具体地, 我们把 1 号点拆成两个点, 一个作为源点只连向 $\mathcal{A}$ 中的点, 另一个作为汇点只被 $\mathcal{B}$ 中的点连向.

然后这里需要一个二进制拆分的技巧: 在与 1 相邻的那些点中, 每次考虑它们二进制下的第 k 位, 将这一位为 0 的放入 $\mathcal{A}$, 为 1 的放入 $\mathcal{B}$, 那么只需 $\log N$ 次, 我们便可以考虑到每一对.

[Codeforces 346D] Robot Control

有一个 N 个点, M 条边的有向图, 初始有一个机器人在 1 号点. 每个时刻, 这个机器人会随机选择一条从该点出发地边并通过. 当机器人到达点 N 时, 它就会自动关闭.

然而这个机器人如果在某个时刻到达自己曾经到过的点的话, 它就会爆炸. 因此, 你决定对机器人实施一些命令, 让它在某些时候按照规定的边走, 而非随机选择.

问对机器人最少使用多少条命令可以让它安全到达点 N.

$N, M \leq 10^6$

首先可以无视掉“不能到达曾经到过的点”的限制, 因为最优答案一定不会存在这种情况.

然后列出 DP 方程:

$$dp(u) = \min_{(u,v)}\{\min\{dp(v)\} + 1, \max\{dp(v)\}\}$$

这个方程可以直接 BFS 计算, 相当于一个边权为 0 或 1 的最短路.

时间复杂度 $O(N + M)$.

[Codeforces 806D] Perishable Roads

给你一个 n 个点的无向完全图，每条边有正边权．对于一个生成有根树，定义其权值为每个点到根的路径上的最小权值之和．对于每个点，求出以它为根的权值最小的生成树．

$n \leq 2000$

性质 1:

一定存在一个最优方案是一条链.

性质 2:

每个最优方案中至少包括一条权值最小的边.

性质 3:

一定存在一个链的最优方案, 使得在根到权值最小的边这一段路径上, 除最后一条边外, 边权严格递减.

不妨将所有边减去最小值, 那么最小的边权将会变为 0, 也就是说我们只需要找到一条根到某条 0 的路径使得其权值最小即可.

如果答案一定满足边权严格递减, 那么直接求最短路. 倒过来考虑, 那么最后一条边会贡献两次, 直接从每个点选一个权值最小的出边, 将这个边权乘 2 作为初始距离, 跑多源最短路即可.

时间复杂度 $O(n^2)$.

常见模型
定义

二分图最小点覆盖
选择最少的点，使得所有边至少有一个端点在集合里.

二分图最小边覆盖
选择最少的边，使得所有点至少与一条边邻接.

二分图最大独立集
选择最多的点，使得任意两被选点不相邻.
(最大团我就不提了，就是补图的最大独立集)

DAG 最大权闭合子图
每个点有点权，可正可负；一条边 (u, v) 代表选 u 就必须选 v.
选择一些点使点权和最大.

DAG 最小路径覆盖

一条路径定义为一个点的序列，对于相邻两个元素，在原图中有边相连.

选择最少的路径，使得每个点恰好被覆盖一次.

DAG 最小链覆盖

一条链定义为一个点集，对于任意两个元素 (u, v)，要么 u 能走到 v，要么 v 能走到 u.

选择最少的链，使得每个点恰好被覆盖一次.

DAG 最长反链

一条反链定义为一个点集，对于任意两个元素 (u, v)，u 不能走到 v，v 也不能走到 u.

找出一个最大的反链.

常见模型

结论

定义 $|\mathcal{V}|$ 为二分图的点数, M 为二分图的最大匹配数.

二分图最小点覆盖 $= M$
证明: 最小割最大流定理.

二分图最小边覆盖 $= |\mathcal{V}| - M$
证明: 考虑将边分类: 贡献为 2 的边和贡献为 1 的边; 再将贡献为 2 的边和最大匹配联系起来.

二分图最大独立集 $= |\mathcal{V}| - M$
证明: 独立集的补集是点覆盖集.

DAG 最大权闭合子图
网络流建模. 将源点连向正权的点, 容量为点权; 将负权的点连向汇点, 容量为点权的绝对值; 原图中的边照原样连, 容量为 $+\infty$.
最终答案为正权的和减去最大流.
证明: 一开始选择所有正权点, 再利用最小割花费最小代价使解合法化.

常见模型

结论

DAG 最小路径覆盖

二分图建模. 将每个点 u 拆成两个点 u_1, u_2, 那么原图中的一条边 (u, v) 对应新图中的一条边 (u_1, v_2).

答案为点数减去最大匹配数. (注意这里的点数不是二分图的点数)

DAG 最小链覆盖

先用 floyd 做一遍传递闭包, 然后做 DAG 上的最小路径覆盖即可.

DAG 最长反链 $=$ 最小链覆盖数

Dilworth 定理.

[Codeforces 590E] Birthday

给你 N 个字符串，要你找出一个集合，使得这个集合中每个元素均不是另外的元素的字串，并且这个集合的大小最大.

要求输出方案.

$N \leq 750$，字符总数不超过 10^7

首先用 AC 自动机处理出一个 DAG，要求出这个 DAG 上的最长反链. 注意这里还要求方案.

首先对 DAG 做传递闭包, 然后将 DAG 上每个点拆成两个, 再从左向右连相应的边, 形成的二分图的最大独立集中, 所有原图中的点满足对应二分图中左边和右边的点都在最大独立集中的点形成了某个最长反链.

证明并不难, 设 M 为二分图的最大匹配数, 首先显然最长反链长度 $\leq$ 最小链覆盖数也就是 $N - M$, 然后我们可以利用最大独立集的大小为 $2N - M$, 而所有原图中的点满足 "对应二分图中左边和右边的点最多只有一个在最大独立集中" 的点的个数不超过 N 个, 所以上面的方法构造的是一条反链, 并且大小 $\geq N - M$.

guard

现在有一副若干条边的二分图，左边有 n 个点，右边有 m 个点，每个点都有一个权值 w_i.

一个合法的点集满足以下两个限制:

1. 选出的点权和大于等于 t
2. 可以从原图中选出若干条边，使得二分图中每个点最多被一条边覆盖，而选出的点要恰好被一条边覆盖 (并不是要求点集内部一定要有完备匹配)

求总方案数.

$$n, m \leq 20, w_i \leq 10^8, t \leq 4 \times 10^8$$

性质: 左边的选取情况和右边的独立. 也就是说, 只要左边和右边的集合满足在整张图中有完备匹配 (根据 Hall 定理) 那么就一定可以.

证明可以先让两边分开找到一个匹配, 合在一起的时候如果出现冲突, 一定存在一条交错轨.

有了这个性质, 两边分别应用 Hall 定理 DP 出合法集合, 最后 two pointers 就行了.

二分图最大匹配关键点/关键边

证明的关键：两个最大匹配之间对称差，只会由若干**长度为偶数**的交替环，交替链组成.

二分图最大匹配关键点

下面只考虑左侧的关键点，右侧的同理.
先求出一个最大匹配. 显然只需考虑匹配点.
从左侧的未盖点出发 DFS，经过的左侧的匹配点**不是**关键点.

二分图最大匹配关键边

先求出一个最大匹配. 显然只需考虑匹配边.
如果一条匹配边的两段存在至少一个非关键点，那么这条边**不是**关键边.
如果一条匹配边在某个偶环，即某个环上 (二分图没有奇环)，那么这条边也**不是**关键边.
实现上，将匹配边和非匹配边定个不同的向，Tarjan 求强连通分量即可.

chess

你现在在玩一种有趣的棋. 这种棋在一个 4×4 的棋盘上进行, 这个棋盘上有一些位置是障碍, 其他位置是空地.

游戏开始时, 有些空地上摆放了一些棋子. 之后每个玩家可以选择两种操作: 将一个棋子移动到相邻的格子中, 或者是移除掉某个棋子. 为了避免游戏无法结束, 每次操作后不允许出现与之前相同的局面.

现在给定一系列局面, 对于每个局面, 你需要判断先手是否一定能够获胜.

不妨观察一下这个博弈图. 首先, 我们可以按照石子个数分层, 显然层与层之间是没有连边的. 那么我们可以预处理出每个点往下走是必胜还是必败.

如是往下走必胜, 那么它就是必胜态. 否则我们把每一层内所有其它状态提取出来 (显然这是个二分图), 然后转化为一个经典问题: 给定一个二分图, 不能经过相同的点, 求必败态.

关键点一定是必胜态, 非关键点一定是必败态.

将 n 个小球放入 m 个篮子里, 每个篮子最多放 5 个小球, 同时放入的小球数量与贡献的关系如下:

0	1	2	3	4	5
3	3	2	2	1	0

给定每个小球可以放入哪些篮子, 求最大贡献之和.

$n, m \leq 100$

类似 “挑战 NPC”.
将每个篮子拆成 7 个点:

```
[1] -> o

[2] -> o
 |
[3] -> o

[4] -> o
 |
[6]

[5] -> o
 |
[7]
```

然后就可以用 WC2016 挑战 NPC 那题一样的做法了.

[POJ 1637] Sightseeing tour

(混合图的欧拉回路)

给你一个既存在有向边, 又存在无向边的图. 问是否存在欧拉回路.

$N \leq 200, M \leq 1000$

难点在于无向边. 考虑每个点的度数限制. 我们先对无向边任意定向, 现在每个点都有一个出度和入度的差; 而我们要求最终每个点出度和入度相等.

考虑网络流. 具体地, 对于我们算出每个点需要将多少条入边改成出边, 或是需要将多少条出边改成入边; 上述两种点一种连 S, 一种连 T 即可; 中间的边即为原图中的无向边. 这些边的容量比较显然, 不再赘述. 满流时有解.

[BZOJ 2324] 营救皮卡丘

给定一个无向图, 边有权. 一开始一共有 K 个人在 0 号点. 每个人可以独立地在边上移动, 移动的代价为边的权值.

到达 i **之前**必须存在至少一个人到达过 $i-1$. 要求到达 N 号点, 并且总代价最小.

$$N \leq 100, M \leq 20000, K \leq 10, 0 \leq w_i \leq 10000$$

首先处理出 $d(i,j)(i<j)$ 表示 i 到 j 只经过编号 $\leq j$ 的点的最短路, 可以用 floyd 求.

考虑费用流.

注意到每个点至少经过一次, 把每个点拆成两个, 类似最小路径覆盖那样建图即可, 唯一的不同是 0 号点和源点的连边容量是 K.

[BZOJ 1937] Mst 最小生成树

给你一个无向图, 边有权. 你可以将任意一些边的权值增大/减小, 每增大/减小 1 将带来 1 的花费.

再给你一个原图的生成树, 要你花费最小的代价, 使得这个生成树成为原图的最小生成树. (不要求唯一)

$N \le 50, M \le 800, 1 \le w_i \le 1000$

如果将生成树上的边当作树边, 其他边当作非树边, 那么显然只会减树边的权值, 加非树边的权值.

设 w_i 为一条边的权值, d_i 为一条边的变化量的绝对值. 设 i 是一个树边, j 是一个非树边, 且 j 覆盖了 i, 则有

$$w_i - d_i \leq w_j + d_j$$

$$d_i + d_j \geq w_i - w_j$$

要求 $\sum d_i$ 最小.

如果把 d_i 看作 KM 算法中的顶标, $w_i - w_j$ 看作一条边, 由于 KM 算法最后顶标和是**最小**的, 那么这个问题就解决了.

[女朋友] gift

有 n 种物品，每种物品有一个价格 a_i，购买了物品 i 需要付出 f_i 的代价.

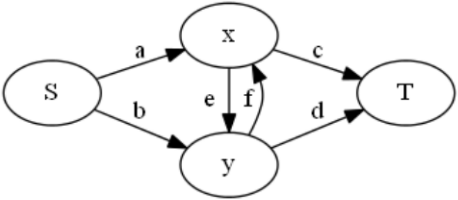
同时，还有 m 对关系 (p_i, q_i, g_i)，表示如果同时购买了物品 p_i 和 q_i，那么就产生 g_i 的收益.

设购买物品一共花费了 F 的代价，产生了 G 的收益，最大化 $\frac{F}{G}$.

$1 \le n \le 4000, 1 \le m \le 200000, a_i, b_i \le 100$

(hint: 网络流点数应当控制在 $O(n)$ 之内)

首先 01 分数规划, 即二分答案 λ 求 $F-\lambda G$ 的最大值.
考虑最小割模型:



$$\begin{cases} a+b & = f_x+f_y \\ c+d & = f_x+f_y-g_{xy} \\ a+d+f & = f_x \\ b+c+e & = f_y \end{cases}$$

最小割:

$$\begin{cases} a+b & = -g_{xy} \\ c+d & = f_x + f_y \\ a+d+f & = f_x - g_{xy} \\ b+c+e & = f_y - g_{xy} \end{cases}$$

求出任意一组解即可:

$$\begin{cases} a = b = -\frac{g_{xy}}{2} \\ c = f_x \\ d = f_y \\ e = f = -\frac{g_{xy}}{2} \end{cases}$$

[BZOJ 1797] 最小割

给你一个有向图, 边有权. 同时给定 S, T 表示源和汇.
要你求出每条边是否

- 至少出现在一种最小割的割集中
- 出现在所有最小割的割集中

$N \leq 4000, M \leq 60000$

先跑一遍网络流. 首先两问的前提都是 (u,v) 满流.

先考虑第一问. 一条边 (u,v) 有可能在最小割中, 当且仅当 u 有可能在 $\mathcal{S}$ 集中, **同时** v 有可能在 $\mathcal{T}$ 集中.

不妨假设, 我们通过连 $(S,u,+\infty),(v,T,+\infty)$, 强制 u 在 $\mathcal{S}$ 集中, v 在 $\mathcal{T}$ 集中, 那么 u 到 v 一定不能有路径.

注意到此时在残量网络中 (u,v) 一定满流, 所以 (u,v) 的反向弧会出现在残量网络中. 所以如果 u 到 v 有路径, 那么将会形成一个强连通分量.

所以只要 u,v 不在同一个强连通分量中, (u,v) 就可能在最小割里.

再考虑第二问. 第二问实际上就是说 u 一定在 $\mathcal{S}$ 集中, 而 v 一定在 $\mathcal{T}$ 集中. 也就是说如果连边 $(u,T,+\infty)$ 强制 u 在 $\mathcal{T}$ 集中, 那么一定会出现新的增广路, 也就是说残量网络中 S 到 u 有路径.

注意到原图 S 到 u 一定有一条增广路, 所以在残量网络中 u 到 S 一定有一条路径. v 同理.

结合上面两个条件, 如果 u 和 S 在一个强连通分量中, v 和 T 在一个强连通分量中, 那么 (u,v) 一定任何一个最小割里.

[Codeforces 280D] k-Maximum Subsequence Sum

给你一列长度为 N, 值小于等于 500 的数列, 支持以下两个操作:

1. 修改一个数的值
2. 对于给定的 l, r, k, 在 $[l, r]$ 中选择不重叠的不超过 k 个的子段, 求最大的和是多少

$N \le 10^5, M \le 10^5, k \le 20$, 操作 2 只有最多 10000 次.

首先考虑如果只询问一次整个序列的答案怎么做.

如果构图构成一条链的样子, 直接费用流就可以了. 考虑优化.

这个费用流做的事情实际上是: 每个点除了权值, 还有一个状态 $0/1$, 每次选中一个权值之和最大的全为状态 0 的区间, 或是一个权值的相反数之和最大的全为状态 1 的区间. 如果这个最大值大于 0, 则对对这个区间的所有权值和状态取反.

用线段树维护即可.

[BZOJ 1061] 志愿者招募

你要为奥运新项目招募一批短期志愿者. 这个项目需要 N 天才能完成, 其中第 i 天至少需要 A_i 个人. 一共有 M 类志愿者可以招募. 其中第 i 类可以从第 S_i 天工作到第 T_i 天, 招募费用是每人 C_i 元. 请你用尽量少的费用招募足够的志愿者.

$$N \leq 1000, M \leq 10000$$

单纯形.

直接从意义入手建模比较困难, 考虑利用流量平衡来建模.

为了叙述方便, 我们首先将输入的 T_i 全部 $+1$, 变成一个左闭右开的区间.

对于第 j 种志愿者, 设 $I_j = [S_j, T_j)$, 表示其能覆盖到的天的集合; x_j 为选的个数.

对于第 i 天, 容易列出不等式:

$$\sum_{i \in I_j} x_j \geq A_i$$

加入松弛变量 y_i:

$$\sum_{i \in I_j} x_j - A_i + y_i = 0$$

对于相邻两项差分 (注意边界情况):

$$\left(\sum_{i\in I_j, i-1\notin I_j} x_j\right) - \left(\sum_{i\notin I_j, i-1\in I_j} x_j\right) + A_{i-1} - A_i + y_i - y_{i-1} = 0$$

$$\left(\sum_{S_j=i} x_j\right) - \left(\sum_{T_j=i} x_j\right) + A_{i-1} - A_i + y_i - yi - 1 = 0$$

$$\left(\sum_{S_j=i} x_j\right) + y_i + A_{i-1} - A_i = \left(\sum_{T_j=i} x_j\right) + y_{i-1}$$

现在就可以利用流量平衡解题了.

一个限制 (即一种志愿者) 对应一条边, 一天对应一个点.

$\sum x_j$ 的部分很好处理, 等价于形如 $(S_j, T_j, +\infty, C_i)$ 的边;

y_i 的部分则等价于 $(i, i-1, +\infty, 0)$ 的边;

$A_{i-1} - A_i$ 由于是常数项, 当 $A_{i-1} - A_i > 0$ 是, 连 $(i, t, A_{i-1} - A_i, 0)$ 的边; 反之, 连 $(s, i, A_i - A_{i-1}, 0)$ 的边.

其实这个常数项本来需要用到流量下界限制的, 但是由于问题十分特殊, 可以直接连边.

在更复杂的问题中, 可能需要用到带上下界的建模方式.

对偶原理

$$\max\{\mathbf{c}^{\mathrm{T}}\mathbf{x} | A\mathbf{x} \leq \mathbf{b}\} = \min\{\mathbf{b}^{\mathrm{T}}\mathbf{y} | A^{\mathrm{T}}\mathbf{y} \geq \mathbf{c}\}$$

假设一个农夫有一块 A 平方千米的农地, 打算种植小麦或大麦, 或是两者依某一比例混合种植. 该农夫只可以使用有限数量的肥料 F 和农药 P, 而单位面积的小麦和大麦都需要不同数量的肥料和农药, 小麦以 (F_1, P_1) 表示, 大麦以 (F_2, P_2) 表示.

设小麦和大麦的售出价格分别为 S_1 和 S_2, 则小麦与大麦的种植面积问题可以表示为以下的线性规划问题:

$$
\begin{array}{lll}
\text{maximize} & S_1x_1 + S_2x_2 & \\
\text{s.t.} & x_1 + x_2 \leq A & \text{(种植面积的限制)} \\
& F_1x_1 + F_2x_2 \leq F & \text{(肥料数量的限制)} \\
& P_1x_1 + P_2x_2 \leq P & \text{(农药数量的限制)} \\
& x_1 \geq 0, x_2 \geq 0 & \text{(不可以栽种负数的面积)}
\end{array}
$$

假如有一个种植园主缺少肥料和农药，他希望同这个农夫谈判付给农夫肥料和农药的价格．构造一个数学模型来研究如何既使得农夫觉得有利可图肯把肥料和农药的资源卖给他，又使得自己支付的金额最少．

假设 y_1, y_2 分别表示每单位肥料和农药的价格，则所支付租金最小的目标函数可以表示为

minimize	$Fy_1 + Py_2$	
s.t.	$F_1y_1 + P_1y_2 \geq S_1$	(使农夫觉得种小麦会亏)
	$F_2y_1 + P_2y_2 \geq S_2$	(与上相似，但为大麦)
	$y_1 \geq 0,\ y_2 \geq 0$	(不可用负数单位金额购买)