网络流

harryzhr

2024 年 12 月 21 日

最大流

不断在残量网络上寻找增广路，直到无法增广为止。此外，在增广的过程中，对于每条边 (u, v)，我们都新建一条反向边 (v, u)，用于退流。

最大流

不断在残量网络上寻找增广路，直到无法增广为止。此外，在增广的过程中，对于每条边 (u, v)，我们都新建一条反向边 (v, u)，用于退流。
那么如何寻找增广路呢?

Dinic 算法

Dinic 算法相比 Edmonds–Karp 算法的优化在于它可以同时增广多条增广路。

- 首先在残量网络上 BFS，给图分层，增广时第 i 层到第 $i + 1$ 层的所有边。
- 在分层图上 DFS，DFS 时采用**当前弧优化**。

Dinic 算法的复杂度为 $O(n^2 m)$，但是需要很特殊的数据才能把 Dinic 卡到这个复杂度。而在实际网络流建模问题中不会出现这么特殊的数据，大可相信 Dinic 跑得很快。

最大流

不断在残量网络上寻找增广路，直到无法增广为止。此外，在增广的过程中，对于每条边 (u, v)，我们都新建一条反向边 (v, u)，用于退流。
那么如何寻找增广路呢?

Dinic 算法

Dinic 算法相比 Edmonds–Karp 算法的优化在于它可以同时增广多条增广路。

- 首先在残量网络上 BFS，给图分层，增广时第 i 层到第 $i + 1$ 层的所有边。
- 在分层图上 DFS，DFS 时采用**当前弧优化**。

Dinic 算法的复杂度为 $O(n^2m)$，但是需要很特殊的数据才能把 Dinic 卡到这个复杂度。而在实际网络流建模问题中不会出现这么特殊的数据，大可相信 Dinic 跑得很快。
还有一些预流推进算法，例如 HLPP，可以把复杂度优化为 $O(n^2\sqrt{m})$，但是实际用处很小，这里就不展开了。

使用 Dinic 做二分图最大匹配

源点 s 连向二分图的左部点，右部点连向 t，然后跑最大流。流量即为最大匹配。
可以证明 Dinic 在二分图中的复杂度是 $O(n\sqrt{m})$。这个复杂度是优于匈牙利算法的。

使用 Dinic 做二分图最大匹配

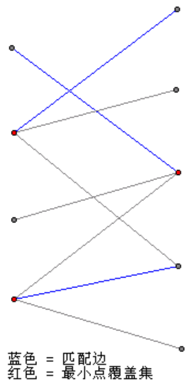
源点 s 连向二分图的左部点，右部点连向 t，然后跑最大流。流量即为最大匹配。
可以证明 Dinic 在二分图中的复杂度是 $O(n\sqrt{m})$。这个复杂度是优于匈牙利算法的。

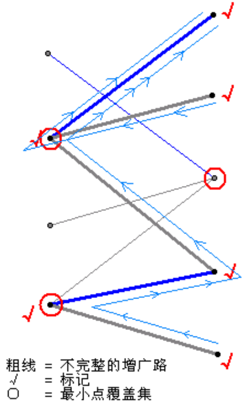
二分图的性质

既然提到了二分图那么就再讲一些二分图的性质吧：

Konig 定理

对于一张二分图，其**最大匹配数**等于其**最小点覆盖数**。
构造如下图所示，从右边的每个非匹配点出发找增广路（一条非匹配边一条匹配边），把经过的所有节点标注出来。





二分图的性质

推论

二分图的最大独立集的大小 $=$ 点数 $n-$ 最大匹配数。
构造取最小点覆盖的补集即可。

二分图的性质

推论

二分图的最大独立集的大小 $=$ 点数 $n-$ 最大匹配数。
构造取最小点覆盖的补集即可。

进一步还可以解决 DAG 的最小路径覆盖问题：

DAG 的最小不相交路径覆盖

二分图的性质

推论

二分图的最大独立集的大小 $=$ 点数 $n-$ 最大匹配数。
构造取最小点覆盖的补集即可。

进一步还可以解决 DAG 的最小路径覆盖问题：

DAG 的最小不相交路径覆盖

将每个点 u 拆成 u_L 和 u_R，作为二分图的左右部点。如果 DAG 有边 $u \to v$，则在二分图上加边 $u_L \to v_R$。
原本每个点作为一条路径，覆盖整个 DAG，每产生一个匹配，就相当于把两条路径拼了起来，路径数量 -1，于是 DAG 的最小不相交路径覆盖的路径数 $=$ 原图的点数 $-$ 二分图的最大匹配数。

DAG 的最小可相交路径覆盖

二分图的性质

推论

二分图的最大独立集的大小 $=$ 点数 $n-$ 最大匹配数。
构造取最小点覆盖的补集即可。

进一步还可以解决 DAG 的最小路径覆盖问题：

DAG 的最小不相交路径覆盖

将每个点 u 拆成 u_L 和 u_R，作为二分图的左右部点。如果 DAG 有边 $u \to v$，则在二分图上加边 $u_L \to v_R$。
原本每个点作为一条路径，覆盖整个 DAG，每产生一个匹配，就相当于把两条路径拼了起来，路径数量 -1，于是 DAG 的最小不相交路径覆盖的路径数 $=$ 原图的点数 $-$ 二分图的最大匹配数。

DAG 的最小可相交路径覆盖

Floyd 求出原图的传递闭包，然后转化为最小不相交路径覆盖。

例题

洛谷 P3511 [POI2010] MOS-Bridges

给定一个图，边有权值且正着走和逆着走有不同权值，在这个图上求一条最大边权最小的欧拉回路，从点 1 出发，要求输出方案。
$n \le 1000, m \le 2000$

最小费用最大流

在保证流量最大的情况下保证贪心地，每次寻找单位费用最小的增广路进行增广，直到图上不存在增广路为止。
因为图中有负权，使用 SPFA，每次找增广路的时间复杂度为 $O(nm)$。设该网络的最大流为 f，则最坏时间复杂度为 $O(nmf)$。

最小费用最大流

在保证流量最大的情况下保证贪心地，每次寻找单位费用最小的增广路进行增广，直到图上不存在增广路为止。
因为图中有负权，使用 SPFA，每次找增广路的时间复杂度为 $O(nm)$。设该网络的最大流为 f，则最坏时间复杂度为 $O(nmf)$。
使用 Primal-Dual 原始对偶算法，类似 Johnson 全源最短路算法地，只使用一次 SPFA，然后给每个点设置势能，把边权变成非负值，然后跑 Dijkstra，而不是每次都跑 SPFA。

例题

洛谷 P2517 [HAOI2010] 订货

某公司估计市场在第 i 个月对某产品的需求量为 U_i，已知在第 i 月该产品的订货单价为 d_i，上个月月底未销完的单位产品要付存贮费用 m，假定第一月月初的库存量为 0，第 n 月月底的库存量也为 0，问如何安排这 n 个月订购计划，才能使成本最低？每月月初订购，订购后产品立即到货，进库并供应市场，于当月被售掉则不必付存贮费。假设仓库容量为 S。

$n \le 50, m \le 10, S \le 10000, U_i \le 10000, d_i \le 100$

例题

洛谷 P3358 最长 k 可重区间集问题

给定数轴上的 n 个开区间，在这些开区间中选出总长度最长的开区间簇，要求 $\forall i \in \mathbb{Z}$，开区间簇中至多有 k 个区间包含 i。
$n \le 500, k \le 3$。

最小割

对于一个网络流图 $G=(V,E)$，其**割**的定义为一种点的划分方式：将所有的点划分为 S 和 $T=V-S$ 两个集合，其中源点 $s\in S$，汇点 $t\in T$。
我们的定义割 (S,T) 的**容量** $c(S,T)$ 表示所有从 S 到 T 的边的容量之和，即 $c(S,T)=\sum_{u\in S,v\in T}c(u,v)$。

最小割

对于一个网络流图 $G=(V,E)$，其**割**的定义为一种点的划分方式：将所有的点划分为 S 和 $T=V-S$ 两个集合，其中源点 $s\in S$，汇点 $t\in T$。
我们的定义割 (S,T) 的**容量** $c(S,T)$ 表示所有从 S 到 T 的边的容量之和，即 $c(S,T)=\sum_{u\in S,v\in T}c(u,v)$。

定理

$$\text{最大流}=\text{最小割}$$

最小割

对于一个网络流图 $G=(V,E)$，其**割**的定义为一种点的划分方式：将所有的点划分为 S 和 $T=V-S$ 两个集合，其中源点 $s\in S$，汇点 $t\in T$。
我们的定义割 (S,T) 的**容量** $c(S,T)$ 表示所有从 S 到 T 的边的容量之和，即 $c(S,T)=\sum_{u\in S,v\in T}c(u,v)$。

定理

$$\text{最大流} = \text{最小割}$$

构造即在跑完最大流的残量网络上，把所有 s 能到的点放进 S，剩下的放进 T。

最大权闭合子图问题

- **闭合子图**：H 是 $G=(V,E)$ 的一个子图，如果 $\forall u \in H, (u,v) \in E$，都有 $v \in E$，则称 H 是 G 的一个闭合子图。
- **最大权闭合子图**：点有点权 a_u，点权有正负，要选出一个闭合子图，最大化它的点权之和。

例题

洛谷 P4001 [ICPC-Beijing 2006] 狼抓兔子

一个 $n \times m$ 的网格图，横向、纵向、斜向（左上-右下）相邻节点均有边连接，边有边权。求边权和最小的一个边集，使得删除边集中的边后左上与右下不连通。

$n, m \leq 1000$

最小割树

洛谷 P4897 【模板】最小割树（Gomory-Hu Tree）

给定一张 n 个点 m 条边的无向图，q 次询问任意两点之间的最小割。
$n \leq 500, m \leq 1500, q \leq 10^5$

上下界网络流

上下界网络流本质是给流量网络的每一条边设置了流量上界 $c(u,v)$ 和流量下界 $b(u,v)$。也就是说，一种可行的流必须满足 $b(u,v) \leq f(u,v) \leq c(u,v)$。同时必须满足除了源点和汇点之外的其余点流量平衡。

无源汇上下界可行流

问题

给定无源汇流量网络 G。询问是否存在一种标定每条边流量的方式，使得每条边流量满足上下界同时每一个点流量平衡。

有源汇上下界可行流

问题

给定有源汇流量网络 G，源点为 s，汇点为 t。询问是否存在一种标定每条边流量的方式，使得每条边流量满足上下界同时每一个点（s, t 除外）流量平衡，且 s 出流量 $\geq$ 入流量，t 入流量 $\geq$ 出流量。

有源汇上下界最大流

问题

给定有源汇流量网络 G。询问是否存在一种标定每条边流量的方式，使得每条边流量满足上下界同时除了源点和汇点每一个点流量平衡。如果存在，询问满足标定的最大流量。

有源汇上下界最小流

问题

给定有源汇流量网络 G。询问是否存在一种标定每条边流量的方式，使得每条边流量满足上下界同时除了源点和汇点每一个点流量平衡。如果存在，询问满足标定的最小流量。

例题

「AHOI2014」支线剧情

给定一个 DAG，边有边权，保证 1 号点可以到达所有节点。你可以出发若干次，每次都从 1 点出发，要求所有边都至少经过 1 次。
最小化你经过的所有边的边权和（重复经过一条边，边权计算多次)。
$n \leq 300$，每个点的出度 ≤ 50，$m \leq 5000$，边权 ≤ 300

网络流题告诉你是啥知识点就没意思了。
所以这部分的题都没有 tag。

例题

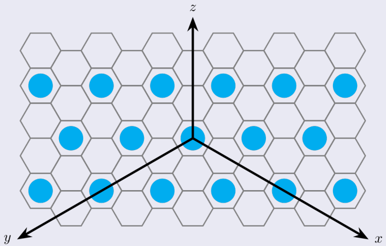
UVA 1306 The K-League

有 n 支队伍进行比赛，每支队伍需要打的比赛数目相同。每场比赛恰好一支队伍胜，另一支败。给出每支队伍目前胜的场数和败的场数，以及每两个队伍还剩下的比赛场数，确定所有可能得冠军的球队（获胜场数最多的得冠军，可以并列）。

$n \leq 50$

洛谷 P5458 [BJOI2016] 水晶

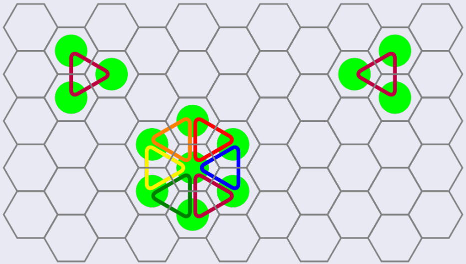
地图由密铺的六边形单元组成，每个单元与其他六个单元相邻。
用坐标 (x, y, z) 描述一个单元的位置，表示从原点开始按如图所示的 x, y, z 方向各走若干步之后到达的地方。有可能有两个坐标描述同一个单元，比如 $(1, 1, 1)$ 和 $(0, 0, 0)$ 描述的都是原点。

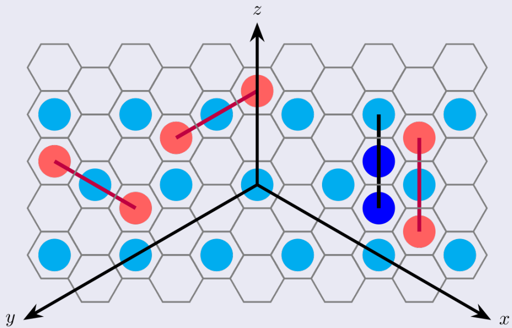


有 N 块水晶位于地图的单元内，第 i 块水晶位于坐标 (x_i, y_i, z_i) 所表示的单元中，并拥有 c_i 的价值，每个单元内部可能会有多块水晶。
地图中，有一些单元安装有能量源。任何满足 $x + y + z$ 是 3 的整数倍的坐标所描述的单元内都安装有能量源。

洛谷 P5458 [BJOI2016] 水晶

有能量源的单元中的水晶价值将会额外增加 10%。你要选择一些水晶，**最大化它们的价值之和**，选出的水晶要满足如下条件：
对于任意三块水晶，它们不能两两相邻地排成一个三角形，也不能三块水晶所在的单元依次相邻地排成一条长度为 2 的直线段，且正中间的单元恰好有能量源。





$1 \le N \le 50000$ $1 \le c_i \le 1000$ $-1000 \le x_i, y_i, z_i \le 1000$

例题

CF724E Goods transportation

小明升任了 CF 国的大总管，他管辖的 n 个城市，编号为 $1..n$ 。每个城市生产了 p_i 个货物，限制最多可以卖掉 s_i 个货物。对于每两个城市 i,j ，如果 $i<j$ ，则可以最多从 i 运送 c 个货物到 j 。注意不能反向运送，却可以在多个城市之间送来送去。现在小明想知道，经过运输后，最多能卖掉多少个货物。

$n \le 10000, c, p_i, s_i \le 10^9$

例题

CF1288F Red-Blue Graph

一张二分图，每个点为红色 / 蓝色 / 无色。现对部分边染为红色（每条花费 r）或蓝色（每条花费 b），使得红色点相连的红色边数严格大于蓝色边数、蓝色点相连的蓝色边数严格大于红色边数，求花费最小的方案（无解输出 -1）。

$n, m, r, b \leq 200$

例题

P2805 [NOI2009] 植物大战僵尸

游戏的地图可以抽象为一个 N 行 M 列的矩阵。在地图的每个位置上都放有一个植物，把位于第 r 行第 c 列的植物记为 $P_{r,c}$。
对每个植物定义 Score 和 Attack 如下：

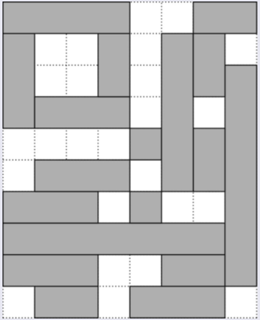
- $\text{Score}(P_{r,c})$：Zombie 击溃植物 $P_{r,c}$ 可获得的能源（可能为负）。
- $\text{Attack}(P_{r,c})$：植物 $P_{r,c}$ 能够保护的位置集合。

Zombies 只能水平移动，若需要对 $P_{r,c}$ 攻击，必须将 $P_{r,M-1}, P_{r,M-2} \cdots P_{r,c+1}$ 先击溃，并移动到位置 (r, c) 才可进行攻击。
Plants 的攻击力是无穷大的，一旦 Zombie 进入某个 Plant 的保护位置，该 Zombie 会被瞬间消灭。(保证 Plant 的保护位置不会包含自身所在位置)。
制定一套 Zombies 的进攻方案，选择进攻哪些植物以及进攻的顺序，从而获得最大的能源收入。
$1 \le N \le 20$，$1 \le M \le 30$，$-10^4 \le \text{Score} \le 10^4$。

例题

CF1404E Bricks

一张 $n \times m$ 的网格图，有些点被染黑。你可以用若干个 $1 \times k$ 或 $k \times 1$ 的长方形纸片去覆盖黑色格子，求最小需要多少个纸片。
不同的长方形纸片的 k 可以不同，k 可以 $= 1$。不可以覆盖白点。
$n, m \leq 400$



例题

「联合省选 2022」学术社区

有 n 个人，总共发出 m 条消息，每条消息可能有一定的价值。每条消息是下面 3 种中的一种：

- 「A：在 B 楼上」，称为楼上型消息，A 为发送人
- 「A：在 B 楼下」，称为楼下型消息，A 为发送人
- 「A：...」，称为学术消息，A 为发送人

价值的定义如下：

- 对于楼上型消息「A：在 B 楼上」，如果它的下一条消息的发送人是 B，那么有 1 的价值，否则没有价值。
- 对于楼下型消息「A：在 B 楼下」，如果它的上一条消息的发送人是 B，那么有 1 的价值，否则没有价值。
- 对于学术消息，没有价值。

构造一个使得价值最大的重排消息的方案。保证每个人都发出了至少 1 条学术消息。

T 组数据。$T \le 100, n \le m \le 77777, \sum m \le 2.5 \times 10^5$。

例题

「SDOI2016」墙上的句子

给定一个 $n \times m$ 的矩阵，矩阵中可能是大写字母，也可能是空格。
每一行可以按照从左到右读和从右到左读两种方式，确定阅读方式后，将这个一行的字符用空格隔开，得到若干单词。每一列也有从上到下和从下到上两种方式，也可以得到若干单词。给每一行和每一列定向后，得到的所有单词的集合称为一个单词表。
有些行和列的阅读方式已经确定，还有一些行和列的阅读方式还不确定。
给剩下未确定的行和列定向，使得最后得到的单词表最小化【s 和 s^R 都在单词表中的不同单词 s 的数量】。
数据保证，每一行（列）从左往右（从上往下）读得到的所有单词同时满足【自己的字典序不小于翻转后的字典序】，或同时满足【自己的字典序不大于翻转后的字典序】。$1 \leq n, m \leq 72$

例题

TopCoder AngelDemonGame

珂珂和花花在一张简单无向图 G 上玩游戏。图 G 拥有 n 个节点，从 $0 \sim n-1$ 编号。你会得到一个描述图 G 的邻接矩阵。珂珂可以选不超过 D 个点对，花花可以选不超过 A 个点对。然后对于图 G 中的一个点对 (i,j):

- 如果珂珂选了 (i,j)，那么新的图当中 (i,j) 之间不存在直接相连的边。
- 否则，如果花花选了 (i,j)，那么新的图当中 (i,j) 之间存在直接相连的边。
- 否则，新图中 (i,j) 之间是否有边取决于原图 G 中 (i,j) 是否有边相连。

在做决策时，珂珂和花花都不知道对方具体的选点对方案是什么。
在新的图当中如果 0 与 $n-1$ 连通，那么花花赢，否则珂珂赢。你需要判断珂珂必胜、花花必胜、或是不属于前面两种情况。这里，A 必胜是指，A 存在一组选点对的方案，使得无论 B 怎么选，A 都会获胜。
$3 \le n \le 50, 2 \le A, D \le \frac{n(n-1)}{2}$

模拟费用流

所谓模拟费用流，就是先把费用流的图建出来，然后在图上通过数据结构/DP 等方式维护增广路，而不是每次都去跑最短路。
和可撤销贪心非常像，有时候其实就是可撤销贪心。

WC2019 专门讲过这个问题，然后当年 NOI 就考了。

这里从当时的 slide《模拟费用流问题》引入。

模拟费用流

所谓模拟费用流，就是先把费用流的图建出来，然后在图上通过数据结构/DP 等方式维护增广路，而不是每次都去跑最短路。
和可撤销贪心非常像，有时候其实就是可撤销贪心。

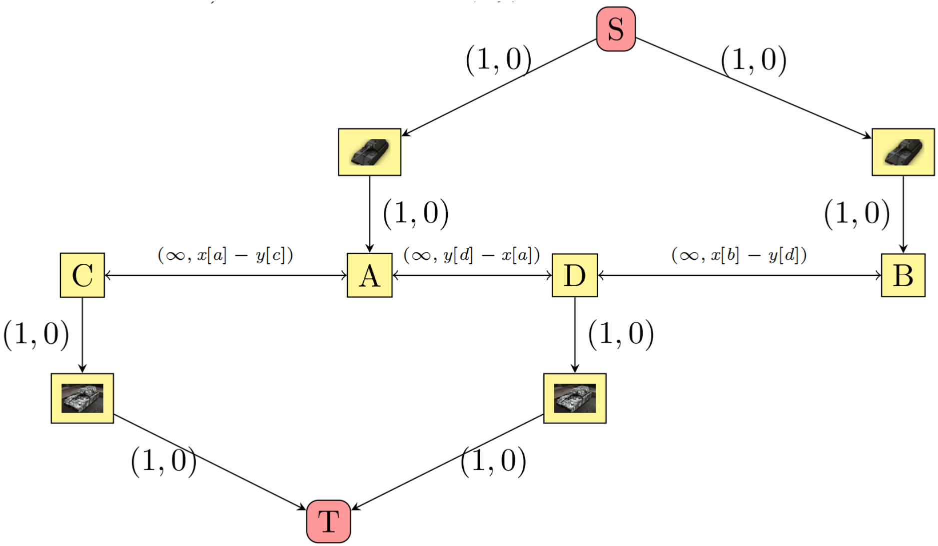
WC2019 专门讲过这个问题，然后当年 NOI 就考了。

这里从当时的 slide《模拟费用流问题》引入。

老鼠进洞问题

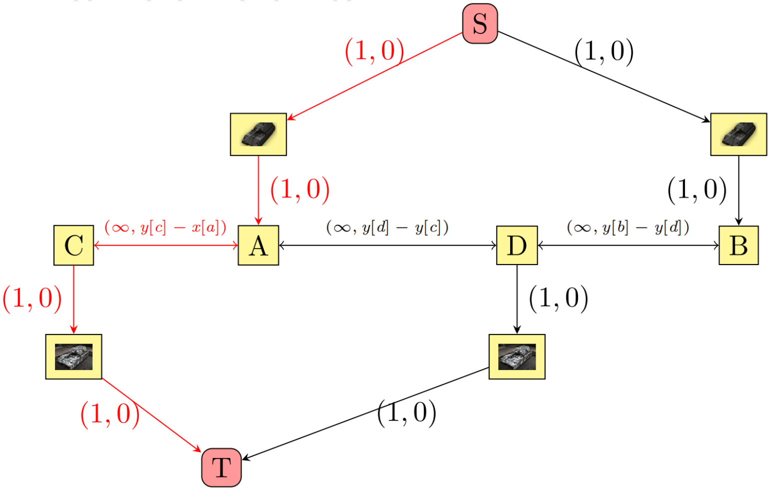
数轴上有 n 只老鼠和 m 个老鼠洞。第 i 只老鼠的坐标为 x_i，第 j 个老鼠洞的坐标为 y_j。
老鼠进一个洞的代价为 $|x_i - y_j|$。
求所有老鼠都进洞的最小总代价（即行走的最小总距离）。

做法很多，下面考虑用一个费用流做法：



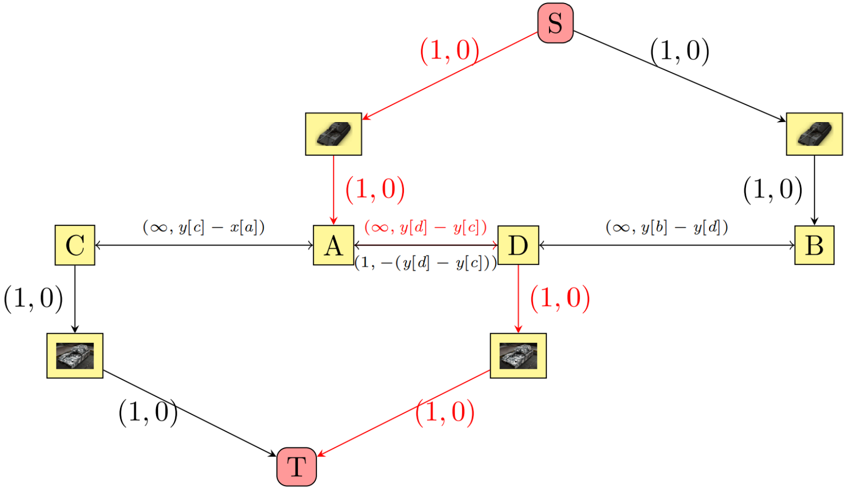
如何快速求费用流?
从左往右考虑每一个老鼠 i，我们找到当前还剩下的洞中左边最近的 L_i 和右边最近的 R_i。
然后分析老鼠到两个洞的距离。

如果 $x_i - y_{L_i} \le y_{R_i} - x_i$，则贪心地匹配左边。



这部分产生的反向边对后面的增广没有影响，可以用栈维护左边还有哪些洞未被匹配。

如果 $x_i - y_{L_i} > y_{R_i} - x_i$，则贪心地匹配右边。



在后一种情况中，反向边是有作用的。
当扫到位于坐标 q 的洞时，如果 $p \to q$ 有一条反向弧，等价于把左边第一个洞的坐标往右移动 $2(q-p)$，即后面的老鼠再匹配左边的洞时，原本需要经过的边的边权是 $q-p$，但现在多了一条边权为 $-(q-p)$ 的可以走，但是这条边只能走一次（退流）。
维护标记即可。复杂度 $O(n \log n)$，瓶颈在排序。

还有一个做法

刚才的做法是每次加上面的边。现在我们考虑从左往右扫，上下的边一起加，只考虑左边的增广。
维护两个堆 Q_1 和 Q_2，

- 对于老鼠 x_i：它要和左边某个洞匹配，代价为 $x_i - y_j$，Q_1 维护剩下的洞中的 $-y_j$ 的最小值。
- 对于一个洞 y_j：它可以“取代”之前某个已经匹配的洞，即找到一个从 T 出发的负环。假设之前某个洞 y_k 与 x_i 匹配，设原本的代价为 w，现在倒过来走这条边，然后再从 x_i 走到 y_k，负环的边权和为 $-w + (y_j - x_i) = y_j + (-w - x_i)$。于是 Q_2 维护所有匹配的 $-w - x_i$ 的最小值即可。取代了 y_k 之后，后面的洞再往左匹配 y_k 时，经过 (x_i, y_j) 这段时边权为负。所以要往 Q_1 里 push 一个 $-y_j - (y_j - x_i) + w = -2y_j + (w + x_i)$。

复杂度 $O(n \log n)$。

例题

「NOI2019」序列

给定两个长度为 n 的正整数序列 $\{a_i\}$ 与 $\{b_i\}$，序列的下标为 $1, 2, \cdots, n$。现在你需要分别对两个序列各指定**恰好** K 个下标，要求**至少**有 L 个下标在两个序列中都被指定，使得这 $2K$ 个下标在序列中对应的元素的总和**最大**。
形式化地说，你需要确定两个长度为 K 的序列 $\{c_i\}, \{d_i\}$，其中 $1 \le c_1 < c_2 < \cdots < c_K \le n, 1 \le d_1 < d_2 < \cdots < d_K \le n$。
并要求 $|\{c_1, c_2, \cdots, c_K\} \cap \{d_1, d_2, \cdots, d_K\}| \ge L$。
目标是最大化

$$\sum_{i=1}^{K} a_{c_i} + \sum_{i=1}^{K} b_{d_i}$$

$T \le 10, 1 \le \sum n \le 10^6, 1 \le L \le K \le n \le 2 \times 10^5, 1 \le a_i, b_i \le 10^9$。

题单

「SHOI2003」吃豆豆
P4553 80 人环游世界
「NOI2008」志愿者招募
「NEERC2016」Mole Tunnels
「SNOI2019」通信
「联合省选 2020 A」魔法商店
「NOI2015」小园丁与老司机

完 结 撒 花