网络流之建图技巧

Appleblue17

2024.7.21

Table of Contents

Definition (网络)

网络是一个特殊的有向图 $G=(V,E)$，每条边 $(u,v)\in E$ 具有容量 $c(u,v)\geq 0$（若 $(u,v)\notin E$ 认为 $c(u,v)=0$）。图中唯一没有入度的点称为源点 s，唯一没有出度的点称为汇点 t。

Definition (流函数)

流函数 $f: V\times V\to \mathbb{R}$，满足：

- 容量限制：$\forall (u,v)\in E, 0\leq f(u,v)\leq c(u,v)$.
- 斜对称：$\forall u,v\in V, f(v,u)=-f(u,v)$
- 流守恒：$\forall u\in V\setminus\{s,t\}, \sum\limits_{(u,v)\in E} f(u,v)=\sum\limits_{(v,u)\in E} f(v,u)$.

Definition (流量)

网络的流量为 $|f|=\sum\limits_{(s,u)\in E} f(s,u)=\sum\limits_{(u,t)\in E} f(u,t)$。

最大流问题

Problem (最大流问题)

给定网络 $G=(V,E)$，求流函数 f 使得流量最大。

最大流问题

Problem (最大流问题)

给定网络 $G=(V,E)$，求流函数 f 使得流量最大。

Ford–Fulkerson 增广

对于边 $(u,v)\in E$，新增一条反向边 (v,u)，容量为 $c(v,u)=-c(u,v)$。
剩余容量：对于所有边 (u,v)，**剩余容量为** $c_f(u,v)=|c(u,v)-f(u,v)|$。
残量网络：V 与所有剩余容量不为 0 的边构成的子图称为**残量网络**，即 $G_f=(V,E_f)$，其中 $E_f=\{(u,v)\mid c(u,v)\neq 0\}$。
增广路：残量网络中一条从 s 到 t 的简单路径。

Ford–Fulkerson 增广方法：每次寻找一条增广路并让增广路上的每条边的流量增加，则整个网络的流量也会增加同样的量。不断重复直到残量网络中源汇不再联通。
由于每次总流量一定会增大，故在重复有限次后一定能够结束。
正确性将通过下面的最大流最小割定理证明。

割

Definition (割)

对于点集 V 的一组划分 $\{S, T\}$，若满足 $s \in S, t \in T$，则称 $\{S, T\}$ 为一组割。

Definition (割边)

对于点集 V 的一组割 $\{S, T\}$，所有满足 $u \in S, v \in T$ 的边 $(u, v) \in E$ 称为割边。

Definition (割的容量)

对于点集 V 的一组割 $\{S, T\}$，其容量为所有割边的容量之和，即：

$$c(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v)$$

最大流最小割定理

Problem (最小割问题)

给定网络 $G = (V, E)$，求一组割 (S, T) 使得其容量 $c(S, T)$ 最小。

Theorem (最大流最小割定理)

对于同一个网络，最大流问题与最小割问题的答案相同，即**最大流等于最小割**。

Proof.

先证明最大流小于等于最小割。对于任意的流 f 与一组割 $\{S, T\}$，有：

$$
\begin{aligned}
|f| &= \sum_{u\in S}\Big(\sum_{(u,v)\in E} f(u,v) - \sum_{(v,u)\in E} f(v,u)\Big) \\
&= \sum_{u\in S}\Big[\sum_{v\in S,(u,v)\in E} f(u,v) + \sum_{v\in T,(u,v)\in E} f(u,v) \\
&\quad - \sum_{v\in S,(v,u)\in E} f(v,u) - \sum_{v\in T,(v,u)\in E} f(v,u)\Big] \\
&= \sum_{u\in S}\sum_{v\in T,(u,v)\in E} f(u,v) - \sum_{v\in T,(v,u)\in E} f(v,u) \\
&\le \sum_{u\in S}\sum_{v\in T,(u,v)\in E} f(u,v) \\
&\le c(S,T)
\end{aligned}
$$

取等当且仅当所有割边均满流，所有割边的反向边（若存在）均空流。

最大流最小割定理

Proof.

再证明能够取等。考虑上面提到的 Ford–Fulkerson 增广方法，在增广结束后残量网络中 s 与 t 一定不连通。
设与 s 连通的点集为 S，记 $T = V \setminus S$，有：

- $\forall u \in S, v \in T, f(u,v) = c(u,v)$，否则 $c_f(u,v) > 0$，$v \in T$，矛盾。
- $\forall u \in T, v \in S, f(u,v) = 0$，否则其新构建的反向边有残余流量，$v \in$，矛盾。

故 FF 增广方法构建出的流 f 满足 $|f| = c(S,T)$。而由上一页的证明，$|f| \le |f_{max}| \le c_{min}(S,T) \le c(S,T)$，这意味着不等号全部取等，即 $|f|$ 为最大流，$c(S,T)$ 为最小割。
至此，既完成了最大流最小割定理的证明，又证明了 FF 增广方法的正确性。 □

本质上，最大流与最小割互为对偶问题。

常用网络流算法

求解最大流：

- Edmonds–Karp 算法：单路增广，理论复杂度 $O(nm^2)$。
- Dinic 算法：多路增广，理论复杂度 $O(n^2m)$。

常用网络流算法

求解最大流：

- Edmonds–Karp 算法：单路增广，理论复杂度 $O(nm^2)$。
- Dinic 算法：多路增广，理论复杂度 $O(n^2m)$。

众所周知网络流的实际复杂度是 $O(\text{玄学})$ 或者 $O(\text{能过})$，并且绝大多数题目都不会特意卡实现方法，掌握以上算法基本足够。

常用网络流算法

求解费用流：SPFA+EK，也可以用 Dijkstra 代替 SPFA（需要用 Johnson 算法的势能），用 Dinic 代替 EK（需要判环）。
理论复杂度为 $O(nmf)$。

常用网络流算法

求解费用流：SPFA+EK，也可以用 Dijkstra 代替 SPFA（需要用 Johnson 算法的势能），用 Dinic 代替 EK（需要判环）。
理论复杂度为 $O(nmf)$。

带上下界限制的网络流：
无源汇上下界可行流：建立超级源汇 ss 和 tt，对每条边 (u,v,l,r) 连边 $(u,v,r-l),(ss,v,l),(u,tt,l)$。
有源汇上下界最大流：先连边 $(t,s,+\infty)$，再跑无源汇可行流，最后从 s 到 t 跑最大流。
有源汇上下界最小流：先连边 $(t,s,+\infty)$，再跑无源汇可行流，最后从 t 到 s 跑最大流。
有源汇上下界最小费用流：把 BFS 换成最短路同样做，但是最后不用跑最大流。

Table of Contents

如何识别网络流题？

- 诡异的数据范围：数据范围很小，不过也有数据范围很大的题目。

如何识别网络流题？

- 诡异的数据范围：数据范围很小，不过也有数据范围很大的题目。
- 非常规而复杂的模型：看上去不太可做。

如何识别网络流题？

- 诡异的数据范围：数据范围很小，不过也有数据范围很大的题目。
- 非常规而复杂的模型：看上去不太可做。
- 线性规划问题：网络流问题本质上是一类线性规划问题。

如何识别网络流题？

- 诡异的数据范围：数据范围很小，不过也有数据范围很大的题目。
- 非常规而复杂的模型：看上去不太可做。
- 线性规划问题：网络流问题本质上是一类线性规划问题。
- 二分图匹配问题：转化为二分图匹配再用网络流求解。

如何识别网络流题？

- 诡异的数据范围：数据范围很小，不过也有数据范围很大的题目。
- 非常规而复杂的模型：看上去不太可做。
- 线性规划问题：网络流问题本质上是一类线性规划问题。
- 二分图匹配问题：转化为二分图匹配再用网络流求解。
- 熟悉的模型：这个套路我见过！

模型一 就是网络流

就是网络流，题目怎么说就怎么建。

「网络流 24 题」餐巾计划问题

在连续的 N 天内，餐厅在第 i 天需要 r_i 块干净的餐巾。干净的餐巾有三种获得方式：

- 购买新餐巾，每块费用为 p。
- 快洗旧餐巾，需 m 天，每块费用 f。
- 慢洗旧餐巾，需 n 天，每块费用 s。

求 N 天内的最小总花费。

数据范围：$1 \le N \le 2 \times 10^3$，$1 \le m < n \le N$，$1 \le r_i \le 10^7$，$1 \le p, f, s \le 10^4$，$s < f$。

「网络流 24 题」餐巾计划问题

按时间分层，对每天建两个点 t, t'，分别表示新旧餐巾。
为了保证每天一定使用 r_i 条新餐巾，先收掉新餐巾（从新餐巾向汇点连边），再提供旧餐巾（从源点向旧餐巾连边），要求满流。
对于快洗，建边 $(t, (t+m)', +\infty, f)$，慢洗同理，跑最大流最小费用流即可。

模型二 二分图匹配

不带权与带权的二分图匹配都可以用网络流求解。
不带权的二分图匹配复杂度为 $O(m\min\{m^{1/2}, n^{2/3}\})$。
先将问题转化为二分图匹配，再用网络流求解。

「网络流 24 题」最小路径覆盖问题

给定有向无环图 G，求最小不交路径覆盖，即用最少的路径覆盖图中的所有点，使每个点恰好被一条路径覆盖。需要给出方案。

数据范围：$1 \le n \le 150$，$1 \le m \le 6000$。

「网络流 24 题」最小路径覆盖问题

路径条数 $=$ 点数 - 路径上边数。问题转化为求路径上边数的最大值。
（如果是无向图的话）按拓扑排序给每条边定向，显然每条路径上的边都是同向的。
每个点只能作为一个入点与一个出点，而每条边就相当于一个匹配。
于是将每个点 u 拆成入点 u 与出点 u'，对于原图中每条边 (u, v) 建边 (u, v')，跑二分图匹配即可。
复杂度 $O(m\sqrt{m})$。

模型三 最小割

将题目建模为最小割模型，再转化为最大流求解。
希望代价最小 $\Rightarrow$ 求最小割 $\Rightarrow$ 求最大流
“禁止”条件：转化为 $+\infty$ 的代价。

模型三 最小割

将题目建模为最小割模型，再转化为最大流求解。
希望代价最小 $\Rightarrow$ 求最小割 $\Rightarrow$ 求最大流
“禁止”条件：转化为 $+\infty$ 的代价。
由此引申出多种模型：集合划分，最大权闭合子图，切糕模型。

模型三 最小割 - 集合划分

Problem

有 n 个物品构成集合 S，要求将 S 划分为 S, T 两个集合，并有如下规则：

1. 若第 i 个物品不在 S 中，则产生 $a_i(a_i > 0)$ 的代价。
2. 若第 i 个物品不在 T 中，则产生 $b_i(b_i > 0)$ 的代价。
3. 给出若干三元组 $(u_t, v_t, w_t)(w_t > 0)$，若物品 $u_t \in S, v_t \in T$（或 $u_t \in T, v_t \in S$）则产生 w_t 的代价。
4. 给出若干二元组 $(S_t, w_t)(w_t > 0)$，其中 S_t 为 S 的非空子集，若 S_t 中的物品不全在 S（或 T）内则产生 w_t 的代价。
5. 给出若干三元组 $(u_t, S_t, w_t)(w_t > 0)$，其中 S_t 为 $S \setminus \{u_t\}$ 的非空子集，若 $u_t \in S$ 而 S_t 中的物品不全在 S 内则产生 w_t 的代价。

模型三 最小割 - 集合划分

Solution

对第 i 个物品建点 i，与 s 和 t 各连一条边。
若割掉与 t 相连的边表示在 S 中，割掉与 s 相连的边表示在 T 中。

1. 建边 (s, i, a_i)。
2. 建边 (i, t, b_i)。
3. 建边 (u_t, v_t, w_t)（或 (v_t, u_t, w_t)）。
4. 建立虚点 x，连边 (s, x, w_t)，对所有 $x \in S_t$ 连边 $(x, u, +\infty)$。
5. 建立虚点 x，连边 (u_t, x, w_t)，对所有 $x \in S_t$ 连边 $(x, u, +\infty)$。实际上包含了规则 4。

容易证明最小割方案中每个物品恰属于一个集合：假设某个物品 u 两边都被割掉，若 $u \in S$，则从割边中删去 (s, u) 会得到更小的割，矛盾；$u \in T$ 同理。
对于**二选一**的情况，代价与贡献可以互相转化：贡献 w 可以转化为预先加上 W，代价为 $W - w$。

「网络流 24 题」方格取数问题

给定 n 行 m 列的方格图，每个格子有权值 $a_{i,j}$。选择一些方格使得任意两个被选择的方格没有公共边，求选中的格子权值之和的最大值。

数据范围：$1 \leq n, m \leq 100$，$1 \leq a_{i,j} \leq 10^5$。

「网络流 24 题」方格取数问题

记集合 S 表示选，T 表示不选。

- 对于每个格子 $u=(i,j)$，先预先加上 a_u（假定选）。若不选（不在 S 中），则产生 $a_{i,j}$ 的代价。
- 对于两个相邻的格子 u,v，若 u,v 都选，则产生 $+\infty$ 的代价。

然而第二条规则不在上面的模型中，怎么办？
注意到网格图是二分图，将二分图一侧的点的选择反过来（即属于 S 表示不选，属于 T 表示选）。
记二分图两侧的点集分别为 P，Q，现在规则变为了：

- 先预先加上 $\sum_{i,j} a_{i,j}$。
- 若 $u=(i,j)\in P$，若不选（不在 S 中），则产生 $a_{i,j}$ 的代价。
- 若 $u=(i,j)\in Q$，若不选（不在 T 中），则产生 $a_{i,j}$ 的代价。
- 对于两个相邻的格子 u,v，不妨设 $u\in P, v\in Q$。若 $u\in S, v\in T$，则产生 $+\infty$ 的代价。

现在转化为标准模型了，建边跑最大流即可。

模型三 最小割 - 最大权闭合子图

Problem

给定点带权有向图 $G=(V,E)$，点 u 的权值为 w_u，求点权和最大的闭合子图。
注意 w_u 可以为负值。

闭合子图：G 的子图 $G'=(V',E')$，满足 $\forall u \in V', (u,v) \in G, v \in V'$。

模型三 最小割 - 最大权闭合子图

Solution

记集合 S 表示选，T 表示不选。

- 对于 $w_u \ge 0$，先预先加上 w_u（假定选）。若不选（不在 S 中），则产生 w_u 的代价。
- 对于 $w_u < 0$，若选（不在 T 中），则产生 $-w_u$ 的代价。
- 对于 $(u, v) \in E$，若 u 选而 v 不选，则产生 $+\infty$ 的代价。

对应建图即：

- 对于 $w_u \ge 0$，先预先加上 w_u，建边 (s, u, w_u)。
- 对于 $w_u < 0$，建边 (u, t, w_u)。
- 对于 $(u, v) \in E$，建边 $(u, v, +\infty)$。

用预先加上的权值和减去最大流即为答案。

模型三 最小割 - 切糕模型

Problem

有 n 个变量 $x_i \in [1, m] \bigcap \mathbb{N}$，当 $x_i = j$ 时会产生代价 $w_{i,j}$。
有若干限制 (u, v, k)，表示要求 $x_v \geq x_u + k$。
求代价最小值。
注意 k 可以为负值。

模型三 最小割 - 切糕模型

Solution

对每个 x_i 建出 $m+1$ 个点，记为点 $(i,j)(j=1,2,\cdots,m+1)$，依次连成一条链。
建边 $(s,(i,1),+\infty),((i,j),(i,j+1),w_{i,j}),((i,m+1),t,+\infty)$，割掉 $((i,j),(i,j+1))$ 就表示选择 $w_i=j$。
对于限制 (u,v,k)，对于每个 j 建边 $((u,j),(v,j+k),+\infty)$。

模型三 最小割 - 切糕模型

Solution

对每个 x_i 建出 $m+1$ 个点，记为点 $(i,j)(j=1,2,\cdots,m+1)$，依次连成一条链。
建边 $(s,(i,1),+\infty),((i,j),(i,j+1),w_{i,j}),((i,m+1),t,+\infty)$，割掉 $((i,j),(i,j+1))$ 就表示选择 $w_i=j$。
对于限制 (u,v,k)，对于每个 j 建边 $((u,j),(v,j+k),+\infty)$。
还有一个问题：怎么防止在同一条链上割掉多条边？
解决方法有很多，这里给出一种：对于 $i=2,3,\cdots,m+1$，建边 $((u,i),(u,i-1),+\infty)$。
这意味着若 $(u,i)\in S$，有 $(u,i-1)\in S$，故必然有且仅有一条割边。
$m=2$ 时的切糕模型就是上面集合划分模型（的一部分）。

模型四 拆点

将一个点拆成两个点并连边，以流量表示选择该点，从而限制选择次数。

LG2045 方格取数加强版

给定 $n \times n$ 网格图，每个格子有权值 $w_{i,j}$。现在行走 k 次，每次从 $(1,1)$ 出发，每次可以向右或向下走，最终到达 (n,n)。求所有被经过的格子的权值之和最大值。

数据范围：$1 \leq n \leq 50$，$0 \leq k \leq 10$，$0 \leq w_{i,j} \leq 1000$。

LG2045 方格取数加强版

每个格子第一次经过时可以得到 $w_{i,j}$ 的贡献，之和就没有贡献了。
把每个格子 $u = (i, j)$ 拆成两个点 u, u'，建边 $(u, u', 1, w_u), (u, u', k, 0)$，跑最大费用最大流即可。
能这么建是因为费用流会优先跑大权值的边，这个 Trick 也可以用于凸贡献的建模。

模型五 区间型一面对多面

先建边 $(s,1),(1,2),\cdots,(n-1,n),(n,t)$。
对于 $[l,r)$ 的区间全部加或减的选择，建边 (l,r,w)，以跳过来反向表示区间加或减。

「网络流 24 题」最长 k 可重区间集问题

给定 n 个开区间 (l_i, r_i)，选择若干开区间。
设选出的开区间集合为 S，则要求 $\forall x \in \mathbb{R}, \sum_{(l_i,r_i)\in S}[l_i < x < r_i] \leq k$。
求选出区间的长度之和的最大值。

数据范围：$1 \leq n \leq 500$，$1 \leq k \leq 3$，$1 \leq l_i < r_i \leq 10^5$。

「网络流 24 题」最长 k 可重区间集问题

Lemma

若所有点的覆盖次数均不超过 k，则一定能将 S 划分为 k 个子集 $T_1, T_2, \cdots, T_k$，使得每个子集内的区间两两不交。

Proof.

在数轴上从小往大扫，设现在扫到了 $x = l_{i_1} = l_{i_2} = \cdots = l_{i_t}$。
那么现在还占用着 T_i 的区间就是所有 $l_i < x < r_i$ 的区间（即 $r_i \geq x + 1$）设这些区间有 s 个。
由于覆盖 $y = x + 0.5$ 的点共 $s + t$ 个，故 $s + t \leq k$，即 $s \leq k - t$，一定能把 t 个区间放进空余的集合里。
扫描完成即完成构造，且离散化后时间复杂度为 $O(n)$。

□

「网络流 24 题」最长 k 可重区间集问题

记值域范围 $W = \max r_i$。先建边 $(s, 1), (1, 2), \cdots, (W-1, W), (W, t)$，流量均为 k，费用均为 0。
对于开区间 (l, r)，建边 $(l, r, 1, r-l)$，有流经过表示选择，这样一条流就代表一个 T_i。
跑最大费用最大流即可。

Table of Contents

T1 CF884F Anti-Palindromize

给定长为 n 的字符串 s，将其任意重排得到 t，满足 $\forall i \in [1, n], t_i \neq t_{n+1-i}$。
给定序列 $\{b_i\}$，求 $\sum_{i=1}^{n}[s_i = t_i]b_i$ 的最大值。

数据范围：$2 \leq n \leq 100$，n 为偶数，s 只含小写字母，$1 \leq b_i \leq 100$。

T1 CF884F Anti-Palindromize

首先显然只关心 s 中每种字母的出现次数，记字母 x 的出现次数为 c_x。
这似乎是一个匹配问题，但是既要将 s 与 t 匹配，又要将 t 内部的字符对匹配，不好处理。
换个思路，将 t 中配对的两个字符 i 与 $n+1-i$ 合成一个点 i，与 s 中字符进行“匹配”。
令 x 与 t_i 和 t_{n+1-i} 间都有流量，这时不能相同的限制转化为了流量限制，可以按这个思路建图。
考虑贡献，对于边 $(x,i,1,w)$：

$$w=\begin{cases}\max\{b_i,b_{n+1-i}\}, & x=s_i=s_{n+1-i}\\ b_i, & x=s_i\neq s_{n+1-i}\\ b_{n+1-i}, & x=s_{n+1-i}\neq s_i\\ 0, & x\neq s_i,x\neq s_{n+1-i}\end{cases}$$

建图：$(s,x,c_x,0)$，$(x,i,1,w)$，$(i,t,2,0)$，跑最大费用最大流即可。

T2 ARC107F Sum of Abs

给出一个 n 个点和 m 条边的简单无向图，每个点有两个权值 a_i 和 b_i。可以以 a_i 的代价删除第 i 个节点以及与这个点相连的边。
一个极大连通块的权值定义为连通块中所有点的 b_i 之和的绝对值。求所有极大连通块权值之和减去代价和的最大值。

数据范围：$1 \le n, m \le 300$，$1 \le a_i \le 10^6$，$-10^6 \le b_i \le 10^6$。

T2 ARC107F Sum of Abs

拆绝对值，于是每个未被删除的点 u 最终的贡献是 b_u 或 $-b_u$。
将每个点拆成两个点 u, u'，分别表示贡献取 b_u 与 $-b_u$。
这是一个集合划分类的问题，考虑用最小割模型。设 S 表示保留，T 表示删去。

- 若删去（不在 S 中），则产生 a_u 的代价。
- u, u' 不能同时保留（在 S 中）。
- 对于 $(u, v) \in E$，u, v' 以及 u', v 不能同时保留。

后两个规则不在模型中，但将所有 u' 的定义反转就可以转化为标准模型。
建图跑最大流即可。

T3 AGC038F Two Permutations

给定两个长为 n 的排列 P, Q，构造两个排列 A 和 B，满足 $\forall i, A_i = i \vee A_i = P_i, B_i = i \vee B_i = Q_i$。求 $\sum_{i=1}^{n} [A_i = B_i]$ 的最大值。

数据范围：$1 \leq n \leq 10^5$。

T3 AGC038F Two Permutations

先考虑排列内的每个循环，那么循环内的每个点 i 要么都不动为 $A_i = i$，要么都变为 $A_i = P_i$； Q 与 B 同理。
这让人联想到最小割模型。对两个排列的每个循环建点，设 S 表示不动，T 表示移位。
而序列上的每个位置就可看作对两边点的一组限制。记位置 i 在 P, Q 中对应的循环分别为 x_i, y_i。

- 先预先加上 n（假定每个位置都能贡献 1）。
- 若 $x_i, y_i \in S$，则必然有 $A_i = B_i$，代价为 1。
- 若 $x_i \in S, y_i \in T$，则代价为 $[i = Q_i]$。
- 若 $x_i \in T, y_i \in S$，则代价为 $[i = P_i]$。
- 若 $x_i \in T, y_i \in T$，则代价为 $[P_i = Q_i]$。

T3 AGC038F Two Permutations

注意到第三、四条规则如果产生了代价，必然有一边的循环长度为 1。此时循环变成了单点，值是确定的，规则就变成了只关于一边的取值。

- 先预先加上 n（假定每个位置都能贡献 1）。
- 若 $i = P_i, i = Q_i$，代价为 1。
- 若 $i = P_i \neq Q_i$，则若 $y_i \in S$ 产生 1 的代价。
- 若 $i = Q_i \neq P_i$，则若 $x_i \in S$ 产生 1 的代价。
- 若 $i \neq P_i, i \neq Q_i$，则若 $x_i, y_i \in S$ 产生 1 的代价。

将 y_i 的定义反转，第五条规则就可以转化为标准模型。跑最大流即可。

T4 ABC347G Grid Coloring 2

给定 $n \times n$ 的矩阵 A，可以将矩阵中任意数量的 0 改为 $1, 2, 3, 4, 5$ 中的任意数。求所有相邻元素差值的平方和的最小值。要求构造方案。

数据范围：$1 \le n \le 20$, $0 \le A_{i,j} \le 5$。

T4 ABC347G Grid Coloring 2

不是必须用到的性质：如果最优解中有 0，将所有 0 改成 1 不会更劣。设改后的矩阵为 B，记 $k=5$。考虑切糕模型，现在问题在于如何刻画代价。
考虑待定解方程，设两点 u,v 间依次连边 $\big((u,i),(v,j),w_{i,j}\big),\big((v,i),(u,j),w_{i,j}\big)$。
如果 $B_u=a,B_v=b$，有：

$$(a-b)^2=\sum_{i=1}^{a}\sum_{j=b+1}^{k}w_{i,j}+\sum_{i=1}^{b}\sum_{j=a+1}^{k}w_{i,j}$$

注意到 $a=b$ 时右式为 0，可以推知 $\forall i<j,w_{i,j}=0$。

T4 ABC347G Grid Coloring 2

$a \neq b$ 时，不妨设 $a < b$，则：

$$(a-b)^2 = \sum_{i=b+1}^{a} \sum_{j=b+1}^{a} w_{i,j} \tag{$*$}$$

由对称性，直接令 $w_{i,i-t} = c_t$，设 $a - b = d$，则 $d^2 = \sum_{t=0}^{d-1}(d-t)c_t$

用数学归纳法可证明 $c_t = \min\{t+1, 2\}$。照模型建图，跑最大流即可。

注意到 $(*)$ 本质上是二维前缀和，所以本质上是二维差分，这要求代价矩阵是蒙日矩阵。

T5 LOJ2146 「SHOI2017」寿司餐厅

有 n 种寿司排成一列，第 i 种寿司的代号为 a_i。
给定 $d_{i,j}(1 \le i \le j \le n)$。每次操作可以将一段区间的寿司全部取下（取下后会立即进行补货，不影响下一次操作)。可以进行任意次操作，设操作的区间依次为 $[l_1, r_1], [l_2, r_2], \cdots$，则获得总美味度为:

$$\sum_{1 \le i \le j \le n} [\exists t, l_t \le i \le j \le r_t] d_{i,j}$$

此外，给定 m, c，对于每个 x，如果吃过 $c > 0$ **种**代号为 x 的寿司，则收费 $mx^2 + cx$；如果没吃过则不收费。
求总美味度减去费用的最大值。

数据范围：$1 \le n \le 100$，$-500 \le d_{i,j} \le 500$，$1 \le a_i \le 1000$，$0 \le m \le 1$。

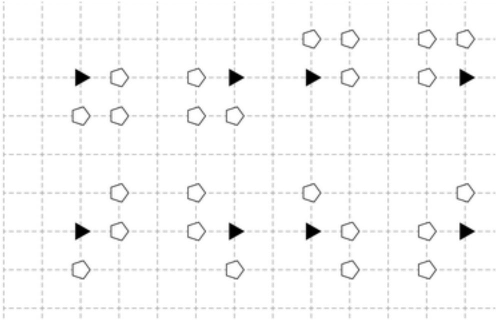
T5 LOJ2146 「SHOI2017」寿司餐厅

容易发现如果 $d_{i,j}$ 被计入贡献，那么 $d_{i-1,j}$ 和 $d_{i,j-1}$ 也会被计入贡献。这是一个最大权闭合子图的模型。如果 $m=0$，那么令每个 $d_{i,i} \to d_{i,i} - a_i$ 附上代价即可。
$m=1$ 时需要描述每种代号的寿司有没有吃过，若吃过有额外代价 mx^2。对每种代号建点，权为 $-mx^2$，让对应的寿司向代号连边，即可转化为最大权闭合子图问题，跑最大流即可。

T6 CF1517G Starry Night Camping

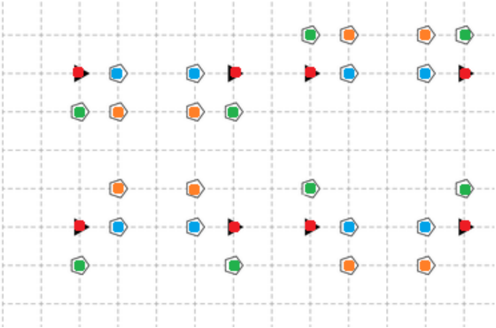
在平面直角坐标系内有 n 个帐篷，帐篷均位于整点且不重叠。第 i 个帐篷的坐标为 (x_i, y_i)，权值为 w_i。
移除一些帐篷，使得不存在一个坐标均为偶数的帐篷，其与八相邻的某三个帐篷组成一条边平行于 x 轴的平行四边形或矩形。
求未被移除的帐篷的权值之和最大值。

数据范围：$1 \leq n \leq 1,000$，$-10^9 \leq x_i, y_i \leq 10^9$，$w_i$：$1 \leq w_i \leq 10^9$。



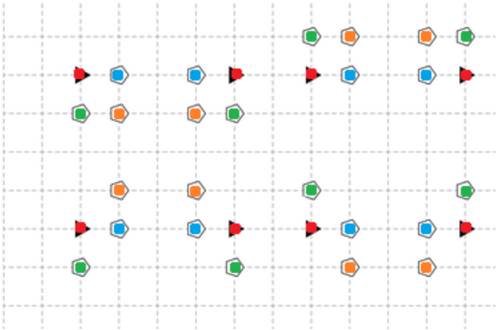
T6 CF1517G Starry Night Camping

“坐标均为偶数”非常可疑，直接四染色：



T6 CF1517G Starry Night Camping

“坐标均为偶数”非常可疑，直接四染色：



Observation

任意一种被禁止的帐篷排布恰好对应一条黄-蓝-红-绿的链。

成功转化为最小割问题。拆点，连边，跑最大流。

T7 CF1250K Projectors

准备进行 n 堂讲课与 m 次研讨会，第 i 堂讲课时间为 $[a_i, b_i)$，第 i 次研讨会时间为 $[p_i, q_i)$。
学校共有 x 台高清投影仪和 y 台普通投影仪。每次讲课或研讨会都需要一台投影仪，讲课必须使用高清投影仪，而研讨会可以使用普通或高清投影仪。
一台投影仪每个时刻只能用在一个地方，且在使用完毕后才会归还。
构造分配方案或报告无解。

数据范围：多组测试，$T \le 300$，
$n, m, x, y \le 300, 1 \le a_i < b_i \le 10^6, 1 \le p_i < q_i \le 10^6$，3s。

T7 CF1250K Projectors

注意到普通投影仪只能在研讨会使用，即需要选择一些研讨会（记作集合 A）使用普通投影仪，而剩下的（记作集合 B）全部使用高清投影仪。由模型五的结论，这等价于在每个时刻，A 中正在进行的不超过 y 个，而 B 中正在进行的不超过 x 个。

考虑用模型五来刻画。先对每场研讨会连边 $(l, r, 1)$。

$$\begin{aligned}
&\text{时刻 } t \text{ 中 } B \text{ 中正在进行场数} \\
&= \text{时刻 } t \text{ 正在进行的总场数} - \text{研讨会对应的边的流量} \\
&= \text{时刻 } t \text{ 正在进行的总场数} - y + \text{边 } (t, t+1) \text{ 流量} \\
&\leq x
\end{aligned}$$

$$\text{边 } (t, t+1) \text{ 流量} \leq x + y - \text{时刻 } t \text{ 正在进行的总场数}$$

注意到右侧如果为负一定无解，于是直接建图跑网络流即可。模型五结论的证明中给出了构造方法。

T8 QOJ1359 [PTZ summer 2020] Setting Maps

给定 n 个点 m 条边的简单有向图以及起点 S 和终点 E。可以标记一些点，标记点 u 的代价为 C_u。最小化使得 S 到 E 的任意一条路径上都至少有 k 个标记点的代价，要求给出方案。

数据范围：$2 \leq n \leq 200$，$1 \leq m \leq 500$，$1 \leq k \leq 5$，$1 \leq C_i \leq 10^7$。

T8 QOJ1359 [PTZ summer 2020] Setting Maps

考虑对于一组标记点方案，怎么判断是否合法。
将每个点 u 拆为入点 u 和出点 u'，如果是标记点就连边权为 1 的边，否则连边权为 0 的边，其余边权均为 0。如果最短路 $\text{dis}(S, E') \geq k$ 就合法。
于是限制为：

- $\forall (u, v) \in E, d_v \leq d_{u'}$。
- $\forall u \in V, d_{u'} \leq d_u + 1$。
- 若 $d'_u = d_u$，则产生 C_u 的代价。
- $d_S = 0$，$d_E = k$。

这是经典切糕模型。设 (u, t) 表示 $d_u = t (t \in [0, k])$，连边 $\big((v, t), (u', t), +\infty\big)$，$\big((u', t), (u, t-1), +\infty\big)$，$\big((u', t), (u, t), C_u\big)$，$\big((u, t), (u, t-1), +\infty\big)$，$\big((u', t), (u', t-1), +\infty\big)$，$\big((S, t), (S, t+1), +\infty\big)(t \geq 1)$，$\big((E', t), (E', t+1), +\infty\big)(t < k)$。
跑最大流即可。

T9 CF843E Maximum Flow

给定一张没有容量的简单网络图（给定源汇），每条边 (u_i, v_i) 有权值 $g_i \in \{0, 1\}$。
请给每条边分配一个正容量 c_i，并构造一组最大流 f，要求：

- 所有 $g_i = 0$ 的边流量为 0；
- 所有 $g_i = 1$ 的边流量不为 0；
- 最小化满流边（即满足 $f_i = c_i$ 的边）的数量。

保证有解，要求构造的方案满足 $1 \le c_i \le 10^9$，$0 \le f_i \le c_i$。

数据范围：$2 \le n \le 100$，$1 \le m \le 1000$，$g_i \in \{0, 1\}$。

T9 CF843E Maximum Flow

最大流最小割定理的证明指出最小割的割边一定满流，所以如果没有 g_i 的限制，答案就是最小割。

考虑 g_i 的限制，相当于要求残量网络上某条边一定有剩余流量。对于 (u_i, v_i)：

- 若 $g_i = 0$，则 (u_i, v_i) 一定有剩余流量，即一定可以从 u_i 走到 v_i。
- 若 $g_i = 1$，则 (v_i, u_i) 一定有剩余流量，即一定可以从 v_i 走到 u_i。

而最大流要求残量网络上源汇不连通，这就变成了一个最小割问题。对于 (u_i, v_i)：

- 若 $g_i = 0$，建边 $(u_i, v_i, +\infty)$。
- 若 $g_i = 1$，建边 $(v_i, u_i, +\infty), (u_i, v_i, 1)$。

跑最大流即可得到答案，同时获得最小割的割边。

至于构造方案，由于要求每条 $g_i = 1$ 的边都要有流量，需要跑有源汇上下界最小流。

T10 QOJ1197 [PTZ winter 2020] Draw in Straight Lines

有一张 $n \times m$ 的网格图，每个格子有黑白两色，初始全为白色。
每次可以进行如下操作之一：

- 将一个格子涂黑或涂白，费用为 c；
- 将同一行或同一列的若干连续的格子全部涂黑或涂白。设操作的格子数量为 l，则费用为 $al + b$。

有如下限制：

- 每个格子至多只能被涂两次。
- 不能将之前被涂白的格子涂黑。

给出目标状态，求让网格图变为目标状态的最小花费。

数据范围：$1 \le n, m \le 40$，$0 \le a, b, c \le 40$，$c \le a + b$。

T10 QOJ1197 [PTZ winter 2020] Draw in Straight Lines

Observation

存在一组最优解，操作顺序依次为：线段涂黑，线段涂白，单点涂。

Proof.

考虑任意一组最优解：
首先，每个格子至多只会被单点涂一次。对于所有的单点涂操作，如果它被后面的涂色覆盖，就可以直接删掉；否则直接移到操作序列的最后。
其次，对于所有线段涂白操作，一定不会被后面的线段涂黑操作覆盖，故可以直接挪到所有线段涂黑操作后面。
这样就得到了操作顺序依次为线段涂黑，线段涂白，单点涂的最优解。 □

T10 QOJ1197 [PTZ winter 2020] Draw in Straight Lines

Observation

最优解中不存在格子被横涂两次或竖涂两次。

Proof.

假设某个格子 x 被横涂两次。

- 若两次涂的是同一种颜色，显然可以合并为一次，且费用严格减小。
- 若两次涂的是不同颜色，设第一次将 $[l_1, r_1]$ 涂黑，第二次将 $[l_2, r_2]$ 涂白。
  - 若 $l_2 \le l_1 \le r_1 \le r_2$，把涂黑操作删掉，费用严格减小。
  - 若 $l_1 \le l_2 \le r_2 \le r_1$，改为 $[l_1, l_2 - 1], [r_2 + 1, r_2]$，费用严格减小。
  - 若 $l_1 \le l_2 \le r_1 \le r_2$，改为 $[l_1, l_2 - 1], [l_2, r_2]$，费用严格减小。
  - 若 $l_2 \le l_1 \le r_2 \le r_1$，改为 $[r_2 + 1, r_2], [l_2, r_2]$，费用严格减小。

与最优解矛盾，故不存在格子被横涂两次或竖涂两次。 □

T10 QOJ1197 [PTZ winter 2020] Draw in Straight Lines

设 $Hb_{i,j}, Vb_{i,j}, Hw_{i,j}, Vw_{i,j}$ 分别表示格子 (i,j) 有没有被横着涂黑，竖着涂黑，横着涂白，竖着涂白。

用最小割模型刻画代价，以竖着涂黑为例（省略剩下的三种）:

- 若 $Hb_{i,j}=1$，则有代价 a。
- 若 $Hb_{i,j}=1, Hb_{i-1,j}=0$，则有代价 b。

单点涂的代价:

- 对于要涂黑的格子，若 $Hw_{i,j}=1$ 或 $Vw_{i,j}=1$ 或 $Hb_{i,j}=Vb_{i,j}=0$，则有代价 c。
- 对于要涂白的格子，若 $Hb_{i,j}=1, Vw_{i,j}=0$ 或 $Vb_{i,j}=1, Hw_{i,j}=0$，则有代价 c。

限制:

- 对于要涂黑的格子，不允许 $Hw_{i,j}=1$ 或 $Vw_{i,j}=1$。
- 对于要涂白的格子，不允许 $Hb_{i,j}=Vb_{i,j}=1$。

注意到有两条规则不在基本模型里，但好巧不巧，将 $Vb_{i,j}$ 和 $Hb_{i,j}$ 的定义反转就可以转化为标准模型，跑最大流即可。

T11 CF1416F Showing Off

有两个 $n \times m$ 的矩阵 A, B，A 中元素为正整数，表示权值；B 中元素为字符 L,R,D,U，表示方向，且 B 中元素代表的方向不会指向矩阵外。
由 A, B 生成矩阵 C，$C_{i,j}$ 表示按照 B 中方向不断行走，所能到达的所有格子的权值之和。
给定 C，构造 A, B 或报告无解。

数据范围：多测，$1 \le T \le 100$，$\sum(n \cdot m) \le 10^5$，$C_{i,j} \in [2, 10^9]$。

T11 CF1416F Showing Off

将方向连成有向边，那么每一个点出度为 1，会形成若干颗内向基环树。

Observation

若两个相邻的格子 u, v 满足 $C_u < C_v$，则不可能存在边 (u, v)。

~~实在是很难注意不到。~~

Observation

存在一组解，只存在长为 2 的环。

Proof.

由于是二分图，所以只会存在偶环。将偶环拆成若干长为 2 的环仍然合法。 $\square$

T11 CF1416F Showing Off

按权值从小到大扫描，设现在只考虑值为 x 的格子，$< x$ 的格子都已经连好了。
注意到若一个值为 x 格子周围有比它小的就可以直接连过去，而如果没有就必须与周围同样为 x 的格子进行匹配形成偶环。
于是现在只剩下一个问题：带强制选择的二分图匹配。

T11 CF1416F Showing Off

Solution 1

先建出图，然后直接暴力上下界网络流。

Solution 2

还有一个优美的构造方法。
先从左侧的必选点向右侧匹配，再从右侧的必选点向左侧匹配，如果失配则无解。
接下来，把两组匹配放在一起。由于每个点度数至多为 2，故会形成若干偶环与链，且边的方向相同
对于偶环或偶链，依次两两配对即可。
对于奇链，从头开始依次两两配对，最后一个点没有出度故不是必选点。

时间复杂度 $O(nm\sqrt{nm})$。

Table of Contents

最大流问题的线性规划形式

为了方便把容量和流量都记成变量形式，即 $c_{u,v}$ 和 $f_{u,v}$。
在原有网络图的基础上新增一条 (t, s) 边以平衡流量，那么原图的流量就是 $f_{t,s}$。
所有变量为：$f_{u,v}((u, v) \in E), f_{t,s}$。

$$
\begin{array}{rll}
\max & f_{t,s} & \\
s.t. & f_{u,v} \leq c_{u,v}, & (u, v) \in E \\
& \displaystyle\sum_{(v,u)\in E} f_{v,u} - \sum_{(u,v)\in E} f_{u,v} = 0, & u \in V \\
& f_{u,v} \geq 0, & (u, v) \in E \\
& f_{t,s} \geq 0 &
\end{array}
$$

最大流问题的线性规划形式

设对偶变量分别为 $d_{u,v}, p_u$，进行对偶：

$$
\begin{aligned}
\min \quad & \sum_{(u,v)\in E} c_{u,v} d_{u,v} \\
s.t. \quad & d_{u,v} + p_v - p_u \geq 0, \quad && (u,v) \in E \\
& p_s - p_t \geq 1 \\
& d_{u,v} \geq 0, && (u,v) \in E \\
& d_{t,s} \geq 0
\end{aligned}
$$

最大流问题的线性规划形式

仔细观察，第一个条件可以改写为 $d_{u,v} \geq p_u - p_v$。
如果设 $p_u = [u \in S], d_{u,v} = [u \in S, v \in T]$，可以发现最小割问题的线性规划模型与最大流几乎完全一样：

$$
\begin{aligned}
\min \quad & \sum_{(u,v)\in E} c_{u,v} d_{u,v} \\
s.t. \quad & d_{u,v} + p_v - p_u \geq 0, \quad & (u,v) \in E \\
& p_s - p_t \geq 1 \\
& d_{u,v} \in \{0,1\}, & (u,v) \in E \\
& d_{t,s} = 0 \\
& p_u \in \{0,1\}
\end{aligned}
$$

易知最大流小于等于最小割。
下面证明最大流等于最小割，即最大流对偶的线性规划中存在一组最优解，使得每个变量都为 0 或 1。

最大流问题的线性规划形式

Proof.

注意到令所有 $p_u \to p_u + C$ 不会影响目标函数的取值，不妨设 $p_t = 0$，$p_s \geq 1$。

对于一组最优解 $(d^*_{u,v}, p^*_u)$，设 $d_{u,v} = d^*_{u,v}, p_u = \min\{p^*_u, 1\}$，下面证明 $(d_{u,v}, p_u)$ 也是一组最优解。

显然第四、五条限制仍然成立，只需考虑第一条限制：

- 若 $p^*_u \leq p^*_v$，则 $p_u \leq p_v$，$d_{u,v} \geq 0 \geq p_u - p_v$。
- 若 $p^*_u > p^*_v > 1$，则 $p_u = p_v = 1$，$d_{u,v} \geq 0 = p_u - p_v$。
- 若 $p^*_u \geq 1 \geq p^*_v$，则 $p_u = 1, p_v = p^*_v$，$d_{u,v} \geq p^*_u - p^*_v \geq p_u - p_v$。

于是这是一组可行解，又由于 $d_{u,v} = d^*_{u,v}$ 故目标函数不变，于是这是一组最优解。

最大流问题的线性规划形式

记最大流对偶线性规划的目标函数最小值为 z。
接下来再证明：存在一组割 $\{S, T\}$，使得 $c(S, T) \leq z$。
令随机变量 $X \sim U(0, 1)$，$S = \{u \mid p_u \geq x\}$，随机变量 $Y = c(S, T)$。
接下来考虑 Y 的期望：

$$
\begin{aligned}
E(Y) &= E\Big(\sum_{(u,v)\in E} [u \in S, v \in T] c_{u,v}\Big) \\
&= \sum_{(u,v)\in E} c_{u,v} E([u \in S, v \in T]) \\
&= \sum_{(u,v)\in E} c_{u,v} P(p_v < X \le p_u) \\
&= \sum_{(u,v)\in E} c_{u,v} \max\{0, p_u - p_v\} \\
&\le \sum_{(u,v)\in E} c_{u,v} (p_u - p_v) \\
&\le \sum_{(u,v)\in E} c_{u,v} d_{u,v} \\
&= z
\end{aligned}
$$

故一定存在 $x \in (0,1)$，使得 $X = x$ 时对应的 $c(S,T) \le z$。
又由于 $z \le c(S,T)$，故 $z = c(S,T)$，即最大流等于最小割。

最小费用最大流问题的线性规划形式

记一组最大流中 $b_u = \sum_{(u,v)\in E} f_{u,v} - \sum_{(v,u)\in E} f_{v,u}$，显然所有最大流的 $\{b_u\}$ 都是一样的。

$$
\begin{array}{rlr}
\min & \displaystyle\sum_{(u,v)\in E} w_{u,v} f_{u,v} & \\
s.t. & f_{u,v} \le c_{u,v}, & (u,v)\in E \\
 & \displaystyle\sum_{(v,u)\in E} f_{v,u} - \sum_{(u,v)\in E} f_{u,v} = -b_u, & u \in V \\
 & f_{u,v} \ge 0, & (u,v)\in E
\end{array}
$$

最小费用最大流问题的线性规划形式

对偶：

$$\begin{aligned}\max \quad & \sum_{(u,v)\in E} c_{u,v}d_{u,v} - \sum_{u\in V} b_u p_u \\ s.t. \quad & d_{u,v} + p_v - p_u \le w_{u,v}, && (u,v)\in E \\ & d_{u,v} \le 0, && (u,v)\in E\end{aligned}$$

令 $z_{u,v} = -d_{u,v}$，改写得：

$$\begin{aligned}\max \quad & -\Big(\sum_{(u,v)\in E} c_{u,v}z_{u,v} + \sum_{u\in V} b_u p_u\Big) \\ s.t. \quad & z_{u,v} \ge p_v - p_u - w_{u,v}, && (u,v)\in E \\ & z_{u,v} \ge 0, && (u,v)\in E\end{aligned}$$

$$\text{Cost} = -\min\{\sum_{u\in V} b_u p_u + \sum_{(u,v)\in E} c_{u,v}\max\{0, p_v - p_u - w_{u,v}\}\}$$

这为费用流模型构建提供了一个新的模型。

最小费用最大流问题的线性规划形式

还有几点需要补充：

1. 一定有整数最优解吗？
由于系数矩阵是全幺模矩阵 [1] ，故这个线性规划模型具有最优整数解特性，即最优解一定为整数向量。

[1] 想了解更多可参考 线性规划 - 整数规划与全幺模矩阵 - Daltao

最小费用最大流问题的线性规划形式

还有几点需要补充：

1. 一定有整数最优解吗？
由于系数矩阵是全幺模矩阵[1]，故这个线性规划模型具有最优整数解特性，即最优解一定为整数向量。

2. 这次代价函数里有 p_u，但对 p_u 还是没有限制，这合理吗？
虽然对 p_u 没有限制，但是 $\sum_u b_u = 0$，故令 $p_u \to p_u + C$ 仍然不会影响目标函数。
由此可见，一定存在一组自然数最优解。

[1] 想了解更多可参考 线性规划 - 整数规划与全幺模矩阵 - Daltao

最小费用最大流问题的线性规划形式

还有几点需要补充：

1. 一定有整数最优解吗？
由于系数矩阵是全幺模矩阵 [1]，故这个线性规划模型具有最优整数解特性，即最优解一定为整数向量。

2. 这次代价函数里有 p_u，但对 p_u 还是没有限制，这合理吗？
虽然对 p_u 没有限制，但是 $\sum_u b_u = 0$，故令 $p_u \to p_u + C$ 仍然不会影响目标函数。
由此可见，一定存在一组自然数最优解。

3. 哪些问题能够通过这种建模解决？
线性规划中的每条限制仅包含两个变量，且不同号；目标函数中所有变量的权值和为 0。

[1] 想了解更多可参考 线性规划 - 整数规划与全幺模矩阵 - Daltao

LG3337 [ZJOI2013] 防守战线

有 n 个位置，在第 i 个位置建塔需要费用 c_i，同一个位置可以建任意多座塔。给定 m 个限制 (L_i, R_i, D_i)，表示区间 $[l_i, r_i]$ 内需至少建 d_i 座塔。求最小总费用。

数据范围：$n \le 1000, m \le 10000, 1 \le l_i \le r_i \le n, c_i, d_i \le 10000$。

Solution 1

设第 i 个位置建了 x_i 个塔。写出线性规划为:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i x_i \\
s.t. \quad & \sum_{i=l_t}^{r_t \geq d_t}, \qquad t \in [1, m] \\
& x_i \geq 0,
\end{aligned}
$$

直接进行对偶:

$$
\begin{aligned}
\max \quad & \sum_{t=1}^{m} d_t y_t \\
s.t. \quad & \sum_{l_t \leq i \leq r_t} y_t \leq c_i, \quad i \in [1, n] \\
& y_t \geq 0
\end{aligned}
$$

这看起来很像区间模型，但是限制是下限而不是上限，故需上下界最大费用流。

Solution 2

设前 i 个位置建了 s_i 个塔。写出线性规划为:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n} c_i(s_i - s_{i-1}) & \\
s.t. \quad & s_i \le s_{i+1}, & i \in [0, n+1] \\
& s_{r_t} - s_{l_t - 1} \ge d_t, & t \in [1, m] \\
& s_i \ge 0 &
\end{aligned}
$$

由于目标函数中变量的权值和为 0,可以将第三条限制去掉。
改写为代价:

$$
\text{Cost} = \sum_{i=0}^{n} s_i(C_i - C_{i+1}) + \infty \max\{0, s_i - s_{i+1}\} + \sum_{t=1}^{m} \infty \max\{0, s_{l_t - 1} - s_{r_t} + d_t\}
$$

建边即:$(s, i, C_i, 0)$,(i, t, C_{i+1}),$(i+1, i, +\infty, 0)$,$(r_t, l_t - 1, +\infty, 0)$。
~~吐槽:竟然敢卡 EK + SPFA!~~

AGC043F Jewelry Box

有 n 间珠宝店，第 i 间商店销售 K_i 种珠宝，每种珠宝有三个属性：重量 $S_{i,j}$，价格 $P_{i,j}$，数量 $C_{i,j}$。
一个珠宝盒包含 n 个珠宝，且满足：

- 不存在两个珠宝来自同一间珠宝店，即珠宝盒恰包含每间店的一件珠宝。
- 给出 m 个形如 (u, v, w) 的限制，表示来自店 v 的珠宝的重量不超过店 u 的重量加上 w。

有 q 次询问，每次询问给出 A，求从珠宝店购买珠宝并组装成 A 个珠宝盒的最小费用，或报告无解。

数据范围：$n, K_i \leq 30, m \leq 50, q \leq 10^5, S_{i,j}, w \leq 10^9, P_{i,j} \leq 30, C_{i,j} \leq 10^{12}, A \leq 3 \times 10^{13}$。

AGC043F Jewelry Box

先对每个珠宝店的珠宝按大小从小到大排序。设 $x_{u,i}$ 表示珠宝店 u 的前 i 种珠宝选择的数量。
显然答案对 A 单调不减，可以改为组至少 A 套珠宝盒。

$$
\begin{array}{lll}
\min & \sum_{u=1}^{n}\sum_{i=1}^{m} P_{u,i}(x_{u,i}, x_{u,i-1}) & \\
s.t. & x_{u,i} \le x_{u,i+1}, & u \in [1,n], i \in [0, K_u - 1 \\
 & x_{u,K_u} - x_{u,0} \ge A, & u \in [1,n] \\
 & x_{u,i} - x_{u,i-1} \le C_{u,i}, & u \in [1,n], i \in [1, K_u] \\
 & \text{设最大的 } j \text{ 使得} S_{v,j} \le S_{u,i} + w, x_{u,i} \le x_{v,j}, & u \in [1,n], i \in [0, K_u]
\end{array}
$$

前面三条限制都好理解，第四条显然是必要条件，下面证明充分性。

AGC043F Jewelry Box

Lemma

若 $x_{u,K_u}=A$，$x_{u,i}\le x_{v,j}$，则能够用选出的珠宝组成 A 个珠宝盒。

Proof.

先给出一组构造方案：将第 u 个店铺选出的第 i 个珠宝放入第 i 个珠宝盒中。
下面证明其合法。考虑反证，假设第 t 个珠宝盒不满足限制 (u,v,w)。
记第 t 个珠宝盒来自店 u,v 的珠宝的标号分别为 p,q，那么
$S_{v,q}>S_{u,p}+w$。
又 $S_{v,j}\le S_{u,p}+w$，故 $q>j$。
而 $x_{v,j}\ge x_{u,p}=t=x_{v,q}$，故 $q\le j$。
推出矛盾，故假设不成立，引理得证。 $\square$

依次建边 $(s,x_{u,i},p_{u,i},0)$，$(x_{u,i},t,p_{u,i+1},0)$，$(x_{u,i},x_{u,i-1},+\infty,0)$，$(x_{u,i-1},x_{u,i},+\infty,C_{u,i})$，$(x_{v,j},x_{u,i},+\infty,0)$，$(x_{u,K_u},x_{u,0},+\infty,-A)$。

AGC043F Jewelry Box

考虑多组询问，实际上 A 改变时只有边 $(x_{u,K_u}, x_{u,0}, +\infty, -A)$ 的权值发生改变。称这种边为“特殊边”。
考虑一组最大流方案，若其流经了 c 条特殊边而非特殊边的费用为 d，那么答案就是 $-\min\{-cA + d\} = \max\{cA - d\}$。
考虑枚举每个 c，求对应的 d 的最小值。
假设确定了每条特殊边的流经次数（比如 $(x_{u,K_u}, x_{u,0})$ 流经 r 次），那么这其实就是上下界网络流。连边 $(s, x_{u,0}, r, 0), (x_{u,K_u}, t, r, 0)$，跑最小费用最大流即可（显然一定能满流）。
但我们并不知道每条特殊边的流经次数。再注意到以下事实：

- 所有 $x_{u,0}$ 都是等价的，可以合成一个点。
- 将 $(x_{u,K_u}, t, r, 0)$ 改为 $(x_{u,K_u}, t, +\infty, 0)$ 不会改变流量与费用。

于是把特殊边删掉，直接连边 $(s, x_{1,0}, r, 0)$，对每个 u 连边 $(x_{u,K_u}, t, +\infty, 0)$ 跑最小费用最大流得到费用 d_c。

AGC043F Jewelry Box

现在只有一条边的边权可变了。先在原图（删掉特殊边）跑一遍，再不断地在这条边上增广，就可以得到所有 d_c。
设第 k 次增广时的最短路为 g_k，那么有 $d_c = \sum_{k=1}^{c} g_k$，且 g_k 单调不减。

$$Ans = \max_c \{cA - \sum_{k=1}^{c} g_k\} = \max_c \{\sum_{k=1}^{c} (A - g_k)\}$$

显然其随 c 先增大后减小，设最大的 c' 使得 $g_{c'} < A$，那么 $c = c'$ 时取到最大值。
于是每次询问二分求值即可。
一次增广的复杂度与费用流相同，又总流量为 $f = O(nkp)$，总时间复杂度为 $O(nmf + q\log f)$。