网络流

济南历城二中 _rqy

2019/02/16

- 需要网络流初步。既然这是 NOI 班（省选班?）那么我就不细讲最基础的板子（比如最大流/最小费用最大流板子）了。

- 需要网络流初步。既然这是 NOI 班（省选班?）那么我就不细讲最基础的板子（比如最大流/最小费用最大流板子）了。
- 似乎没了。

模型：最大流

- 给定一张带权有向图，其中有两个特定的点称之为 **源点**和 **汇点**（以后均以 S 和 T 指代）

模型：最大流

- 给定一张带权有向图，其中有两个特定的点称之为 **源点**和 **汇点**（以后均以 S 和 T 指代）
- 有向边的权值代表了这条边上能够运输多少物品，称为 **容量**。

模型：最大流

- 给定一张带权有向图，其中有两个特定的点称之为 **源点**和 **汇点**（以后均以 S 和 T 指代）
- 有向边的权值代表了这条边上能够运输多少物品，称为 **容量**。
- 求从源点到汇点共可以运输多少物品。

数学模型

- 原图中的每条边 (u, v) 会拆成两条边 (u, v) 和 (v, u)，前者容量为原图边权，后者容量为 0。

数学模型

- 原图中的每条边 (u, v) 会拆成两条边 (u, v) 和 (v, u)，前者容量为原图边权，后者容量为 0。
- 容量限制：$\forall(u, v) \in E, f_{u,v} \leq c_{u,v}$。

数学模型

- 原图中的每条边 (u, v) 会拆成两条边 (u, v) 和 (v, u)，前者容量为原图边权，后者容量为 0。
- 容量限制：$\forall(u, v) \in E, f_{u,v} \leq c_{u,v}$。
- 流量平衡：$\forall u \in V \backslash \{S, T\}, \sum_{(u,v)\in E} f_{u,v} = 0$。

数学模型

- 原图中的每条边 (u, v) 会拆成两条边 (u, v) 和 (v, u)，前者容量为原图边权，后者容量为 0。
- 容量限制：$\forall (u, v) \in E, f_{u,v} \le c_{u,v}$。
- 流量平衡：$\forall u \in V \backslash \{S, T\}, \sum_{(u,v)\in E} f_{u,v} = 0$。
- 斜对称性：$\forall (u, v) \in E, f_{u,v} = -f_{v,u}$。

算法

- Dinic 基本上已经家喻户晓了。ISAP 也有不少人会。
  这里就直接略过（毕竟已经说好不讲模板了）。

算法

- Dinic 基本上已经家喻户晓了。ISAP 也有不少人会。
  这里就直接略过（毕竟已经说好不讲模板了）。
- 另外有时候图中的点也有容量，即流过它的流量不能超过某个数。
  这时候把点 x 拆成两个 $x1$ 和 $x2$，原来 x 的入边都连向 $x1$，出边都从 $x2$ 连出去，然后从 $x1$ 到 $x2$ 连一条容量为点容量的边即可。

算法

- Dinic 基本上已经家喻户晓了。ISAP 也有不少人会。
  这里就直接略过（毕竟已经说好不讲模板了）。
- 另外有时候图中的点也有容量，即流过它的流量不能超过某个数。
  这时候把点 x 拆成两个 $x1$ 和 $x2$，原来 x 的入边都连向 $x1$，出边都从 $x2$ 连出去，然后从 $x1$ 到 $x2$ 连一条容量为点容量的边即可。
- 那么接下来就要讲一些题目了。
  PS：所有例题及练习题的代码（并简要讲解，以注释形式写在代码开头）都会下发。
  PS2：课件中为减少篇幅，所有例题的输入输出格式等细节均没有给出，做题之前请先去看一遍完整题面。

题意

来自 m 个不同单位的代表参加会议并准备就餐。第 i 个单位的代表是为 r_i。

共有 n 个餐桌，第 i 个餐桌可以坐 c_i 人。

要求同一个单位的代表不能在同一个餐桌就餐。请给出一个就餐安排，或者输出无解。

$m \leqslant 150, n \leqslant 170$。

解法

- 既然这节是网络流那么显然我们要用网络流的思想去思考问题。

解法

- 既然这节是网络流那么显然我们要用网络流的思想去思考问题。
- 考虑到“同一个单位的代表不能在同一个餐桌就餐”相当于说“每个单位在每个餐桌最多排一个代表”，可以想到给每个单位建一个点，每个餐桌建一个点，每个单位向每个餐桌建一条容量为 1 的边即可。这里每单位流量就是一个代表。

解法

- 既然这节是网络流那么显然我们要用网络流的思想去思考问题。
- 考虑到“同一个单位的代表不能在同一个餐桌就餐”相当于说“每个单位在每个餐桌最多排一个代表”，可以想到给每个单位建一个点，每个餐桌建一个点，每个单位向每个餐桌建一条容量为 1 的边即可。这里每单位流量就是一个代表。
- 接下来只需要建出源点 S 和汇点 T，并从 S 到每个单位连边，容量为 r_i，每个餐桌到 T 连边，容量为 c_i。
  然后跑最大流，如果流量不等于总人数说明无解，否则输出方案只需要判断每个单位向哪些餐桌连的边满流即可。

解法

- 既然这节是网络流那么显然我们要用网络流的思想去思考问题。
- 考虑到“同一个单位的代表不能在同一个餐桌就餐”相当于说“每个单位在每个餐桌最多排一个代表”，可以想到给每个单位建一个点，每个餐桌建一个点，每个单位向每个餐桌建一条容量为 1 的边即可。这里每单位流量就是一个代表。
- 接下来只需要建出源点 S 和汇点 T，并从 S 到每个单位连边，容量为 r_i，每个餐桌到 T 连边，容量为 c_i。
  然后跑最大流，如果流量不等于总人数说明无解，否则输出方案只需要判断每个单位向哪些餐桌连的边满流即可。
- PS: 本题还有简单的贪心做法，但于本节内容无关，这里不再赘述。

题意

给定一个 n 个点 m 条边的 DAG。

要求用尽量少的路径覆盖所有点，即选择尽量少的路径，使得每个点都恰好在一条路径上。

$n \leqslant 150, m \leqslant 6000$。

解法

- 考虑不相交的路径覆盖中每个点的入度和出度都至多为 1，而且路径条数等于 n - 边数，
  实际上只需要最大化边数即可。而限制就是每个点的出边和入边都最多选择一条。

解法

- 考虑不相交的路径覆盖中每个点的入度和出度都至多为 1，而且路径条数等于 n - 边数，
  实际上只需要最大化边数即可。而限制就是每个点的出边和入边都最多选择一条。
- 可以把每个点 u 拆成两个点 u 和 u'，分别提供入度和出度（也就是二分图匹配）。

解法

- 考虑不相交的路径覆盖中每个点的入度和出度都至多为 1，而且路径条数等于 n - 边数，
实际上只需要最大化边数即可。而限制就是每个点的出边和入边都最多选择一条。
- 可以把每个点 u 拆成两个点 u 和 u'，分别提供入度和出度（也就是二分图匹配）。
- 从源点 S 到每个点 u 连边，容量为 1，从每个点 u' 向 T 连边，容量为 1；原图中如果 u 可以到达 v，就从 u 到 v' 连一条边，容量为 1。

解法

- 考虑不相交的路径覆盖中每个点的入度和出度都至多为 1，而且路径条数等于 n - 边数，
  实际上只需要最大化边数即可。而限制就是每个点的出边和入边都最多选择一条。
- 可以把每个点 u 拆成两个点 u 和 u'，分别提供入度和出度（也就是二分图匹配）。
- 从源点 S 到每个点 u 连边，容量为 1，从每个点 u' 向 T 连边，容量为 1;
  原图中如果 u 可以到达 v，就从 u 到 v' 连一条边，容量为 1。
- n - 最大流就是答案。

例子

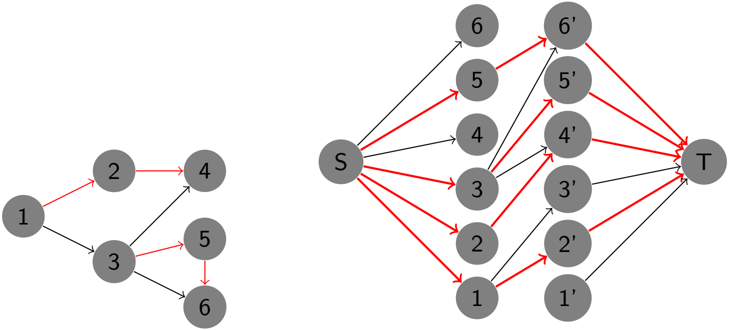

Figure: 一个例子。右边图中所有的边容量均为1，红色边是满流边。

模型：最小割

- 给定一张带权有向图，其中有两个点 S 和 T。

模型：最小割

- 给定一张带权有向图，其中有两个点 S 和 T。
- 要求删掉若干条边，使得不存在从 S 到 T 的路径。

模型：最小割

- 给定一张带权有向图，其中有两个点 S 和 T。
- 要求删掉若干条边，使得不存在从 S 到 T 的路径。
- 最小化删掉的边的边权之和。

模型：最小割

- 给定一张带权有向图，其中有两个点 S 和 T。
- 要求删掉若干条边，使得不存在从 S 到 T 的路径。
- 最小化删掉的边的边权之和。
- 另一种解释方式是：把图上的点划分成两个集合，使得 S 和 T 不在同一个集合中，并最小化从 S 所在集合到 T 所在集合的所有边的边权之和。

算法

- 根据最小割最大流定理，最小割 $=$ 最大流。

算法

- 根据最小割最大流定理，最小割 = 最大流。
- 如何构造一个最小割?

算法

- 根据最小割最大流定理，最小割 $=$ 最大流。
- 如何构造一个最小割?
- 跑完最大流之后，残量网络上 S 能到达的点作为一个集合，S 无法到达的点作为另一个集合。

算法

- 根据最小割最大流定理，最小割 $=$ 最大流。
- 如何构造一个最小割?
- 跑完最大流之后，残量网络上 S 能到达的点作为一个集合，S 无法到达的点作为另一个集合。
- 与最大流一样，还有最小点割，即把割掉边改成割掉点。这时候和最大流的点带容量的做法相同。

题意

有 m 个设备和 n 个实验，每个实验会给定一个设备的集合，表示它需要哪些设备。

再给出每个设备的价格和每个实验的收益，要求买若干设备并选择一些实验来做，最大化净收益。

设备都是可以多次使用的，也就是说每个设备只需要买一次。

$n, m \leqslant 50$。

解法

- 考虑对每个设备和每个实验建一个点。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。
- 我们先假装获得了所有收益，然后减去需要的代价以及实际上不会获得的收益即可。
  考虑所有的限制：

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。
- 我们先假装获得了所有收益，然后减去需要的代价以及实际上不会获得的收益即可。
  考虑所有的限制：
  1. 如果购买某个设备，需要付出其价格：从 S 到这个设备连一条边权为其价格的边，表示割掉这条边才能让它划分到 T 集合（购买）。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。
- 我们先假装获得了所有收益，然后减去需要的代价以及实际上不会获得的收益即可。
  考虑所有的限制：
  1. 如果购买某个设备，需要付出其价格：从 S 到这个设备连一条边权为其价格的边，表示割掉这条边才能让它划分到 T 集合（购买）。
  2. 如果不做某个实验，会损失其收益：从这个实验到 T 连一条边权为其收益的边，割掉这条边才能让它划分到 S 集合（不做实验）。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。
- 我们先假装获得了所有收益，然后减去需要的代价以及实际上不会获得的收益即可。
  考虑所有的限制：
  1. 如果购买某个设备，需要付出其价格：从 S 到这个设备连一条边权为其价格的边，表示割掉这条边才能让它划分到 T 集合（购买）。
  2. 如果不做某个实验，会损失其收益：从这个实验到 T 连一条边权为其收益的边，割掉这条边才能让它划分到 S 集合（不做实验）。
  3. 如果某实验要求某设备，那么不买这个设备就不能做这个实验：从设备代表的点向实验代表的点连边权为 ∞ 的边即可，表示如果设备属于 S 集合（不买）那这个实验也要属于 S 集合（不做）。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。
- 我们先假装获得了所有收益，然后减去需要的代价以及实际上不会获得的收益即可。
  考虑所有的限制：
  1. 如果购买某个设备，需要付出其价格：从 S 到这个设备连一条边权为其价格的边，表示割掉这条边才能让它划分到 T 集合（购买）。
  2. 如果不做某个实验，会损失其收益：从这个实验到 T 连一条边权为其收益的边，割掉这条边才能让它划分到 S 集合（不做实验）。
  3. 如果某实验要求某设备，那么不买这个设备就不能做这个实验：从设备代表的点向实验代表的点连边权为 ∞ 的边即可，表示如果设备属于 S 集合（不买）那这个实验也要属于 S 集合（不做）。
- 这样我们就得到了一张图，用所有实验的收益之和减去它的最小割即可。

解法

- 考虑对每个设备和每个实验建一个点。
- 既然最小割模型中要把所有点划分成两个集合，我们可以指定：
  1. 设备代表的点，划分到 S 所在集合表示不购买，划分到 T 所在集合表示购买。
  2. 实验代表的点，划分到 S 所在集合表示不做实验，划分到 T 所在集合表示做实验。
- 我们先假装获得了所有收益，然后减去需要的代价以及实际上不会获得的收益即可。
  考虑所有的限制：
  1. 如果购买某个设备，需要付出其价格：从 S 到这个设备连一条边权为其价格的边，表示割掉这条边才能让它划分到 T 集合（购买）。
  2. 如果不做某个实验，会损失其收益：从这个实验到 T 连一条边权为其收益的边，割掉这条边才能让它划分到 S 集合（不做实验）。
  3. 如果某实验要求某设备，那么不买这个设备就不能做这个实验：从设备代表的点向实验代表的点连边权为 ∞ 的边即可，表示如果设备属于 S 集合（不买）那这个实验也要属于 S 集合（不做）。
- 这样我们就得到了一张图，用所有实验的收益之和减去它的最小割即可。
- 至于输出方案，只需要构造出最小割即可。

例子

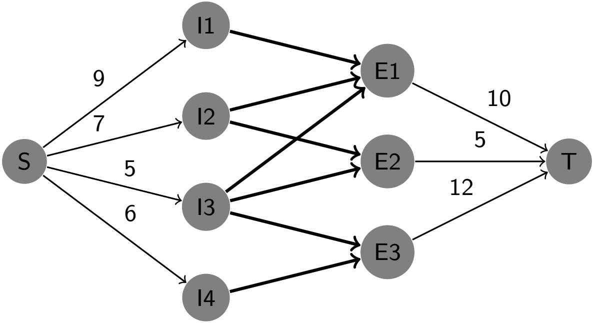

Figure: 一个例子：三个实验的收益分别是 $10, 5, 12$，四种设备的价格分别是 $9, 7, 5, 6$，实验 1 需要设备 $1, 2, 3$，实验 2 需要设备 $2, 3$，实验 3 需要设备 $3, 4$。加粗的边的边权均为 ∞。

例子

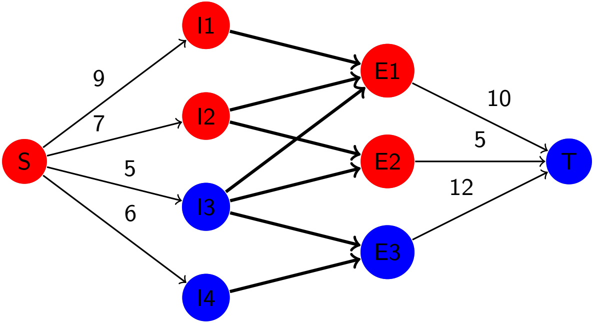

Figure: 一个例子：三个实验的收益分别是 $10,5,12$，四种设备的价格分别是 $9,7,5,6$，实验 1 需要设备 $1,2,3$，实验 2 需要设备 $2,3$，实验 3 需要设备 $3,4$。加粗的边的边权均为 ∞。最小割中实验 3 和设备 $3,4$ 划分到 T 集合，表示做这个实验、使用这两个设备。

本题模型：最大权闭合子图

- 有若干物品，每个物品带权（可以为负），物品间有依赖（形如“选物品 i 则必选物品 j”）。
  最大化选取的物品的权值之和。

本题模型：最大权闭合子图

- 有若干物品，每个物品带权（可以为负），物品间有依赖（形如“选物品 i 则必选物品 j”）。
  最大化选取的物品的权值之和。
- 考虑在最小割里把“与 S 划分在一个集合”和“与 T 划分到一个集合”看成“不选”和“选”。（当然如果反过来也是可以的，建图的时候把边都反过来就好了）。
  那么如果一个物品的权值是正的，就从它到 T 连边，边权是它的权值，表示如果不选它就要损失掉这个权值。
  否则的话就从 S 到它连边，边权是其权值的相反数，表示如果选它就会得到这个权值也就是损失其相反数的收益。

本题模型：最大权闭合子图

- 有若干物品，每个物品带权（可以为负），物品间有依赖（形如“选物品 i 则必选物品 j”）。
  最大化选取的物品的权值之和。
- 考虑在最小割里把“与 S 划分在一个集合”和“与 T 划分到一个集合”看成“不选”和“选”。（当然如果反过来也是可以的，建图的时候把边都反过来就好了）。
  那么如果一个物品的权值是正的，就从它到 T 连边，边权是它的权值，表示如果不选它就要损失掉这个权值。
  否则的话就从 S 到它连边，边权是其权值的相反数，表示如果选它就会得到这个权值也就是损失其相反数的收益。
- 如果物品 i 依赖物品 j，那么就从 j 到 i 连一条边权为 ∞ 的边，表示如果选了 i，即 i 和 T 划分在同一个集合，就必须选 j。

本题模型：最大权闭合子图

- 有若干物品，每个物品带权（可以为负），物品间有依赖（形如“选物品 i 则必选物品 j”）。
  最大化选取的物品的权值之和。
- 考虑在最小割里把“与 S 划分在一个集合”和“与 T 划分到一个集合”看成“不选”和“选”。（当然如果反过来也是可以的，建图的时候把边都反过来就好了）。
  那么如果一个物品的权值是正的，就从它到 T 连边，边权是它的权值，表示如果不选它就要损失掉这个权值。
  否则的话就从 S 到它连边，边权是其权值的相反数，表示如果选它就会得到这个权值也就是损失其相反数的收益。
- 如果物品 i 依赖物品 j，那么就从 j 到 i 连一条边权为 ∞ 的边，表示如果选了 i，即 i 和 T 划分在同一个集合，就必须选 j。
- 最后输出所有正的权值和减去最小割即可。

本题模型：最大权闭合子图

- 有若干物品，每个物品带权（可以为负），物品间有依赖（形如“选物品 i 则必选物品 j”）。
  最大化选取的物品的权值之和。
- 考虑在最小割里把“与 S 划分在一个集合”和“与 T 划分到一个集合”看成“不选”和“选”。（当然如果反过来也是可以的，建图的时候把边都反过来就好了）。
  那么如果一个物品的权值是正的，就从它到 T 连边，边权是它的权值，表示如果不选它就要损失掉这个权值。
  否则的话就从 S 到它连边，边权是其权值的相反数，表示如果选它就会得到这个权值也就是损失其相反数的收益。
- 如果物品 i 依赖物品 j，那么就从 j 到 i 连一条边权为 ∞ 的边，表示如果选了 i，即 i 和 T 划分在同一个集合，就必须选 j。
- 最后输出所有正的权值和减去最小割即可。
- 刚才那道题就是最大权闭合子图的一个例子（每个设备的权值是负的，每个实验的权值是正的）。

题意

$n \times m$ 的网格中每个格子有一个同学。

这些同学要进行文理分科。如果坐在 (i, j) 的学生选文科就会产生 $art_{i,j}$ 的满意值，选理科就会产生 $science_{i,j}$ 的满意值。

此外，如果坐在 (i, j) 的学生与所有和他相邻的学生都选了文科就会产生 $same\_art_{i,j}$ 的满意值，全选理科就会产生 $same\_science_{i,j}$ 的满意值。

要求最大化满意值之和。

$n, m \leqslant 100$。

解法

- 本题虽然不能直接套最大权闭合子图的模型（虽然真的要套也可以），但是基本想法还是一样的。

解法

- 本题虽然不能直接套最大权闭合子图的模型（虽然真的要套也可以），但是基本想法还是一样的。
- 我们对每个学生建一个点。如果这个点和 S 分到一边，说明他选文，否则选理。
  那么类似之前的做法，我们从 S 到它连一条边权为 $art_{i,j}$ 的边，从它到 T 连一条边权为 $science_{i,j}$ 的边。

解法

- 本题虽然不能直接套最大权闭合子图的模型（虽然真的要套也可以），但是基本想法还是一样的。
- 我们对每个学生建一个点。如果这个点和 S 分到一边，说明他选文，否则选理。
  那么类似之前的做法，我们从 S 到它连一条边权为 $art_{i,j}$ 的边，从它到 T 连一条边权为 $science_{i,j}$ 的边。
- 再来考虑 $same\_art$。我们对每个 $same\_art$ 再建一个点，它划分到 S 一边表示可以获得这个收益，否则则不能。
  那么从 S 到它连一条边权为 $same\_art_{i,j}$ 的边即可。而它要求对应的一些学生都要选文（连 S），所以从它向那些学生连一条边权为 ∞ 的边即可。

解法

- 本题虽然不能直接套最大权闭合子图的模型（虽然真的要套也可以），但是基本想法还是一样的。
- 我们对每个学生建一个点。如果这个点和 S 分到一边，说明他选文，否则选理。
  那么类似之前的做法，我们从 S 到它连一条边权为 $art_{i,j}$ 的边，从它到 T 连一条边权为 $science_{i,j}$ 的边。
- 再来考虑 $same\_art$。我们对每个 $same\_art$ 再建一个点，它划分到 S 一边表示可以获得这个收益，否则则不能。
  那么从 S 到它连一条边权为 $same\_art_{i,j}$ 的边即可。而它要求对应的一些学生都要选文（连 S），所以从它向那些学生连一条边权为 ∞ 的边即可。
- $same\_science$ 类似，从那些学生向它连 ∞ 的边，然后从它向 T 连边权为 $same\_science$ 的边即可。

题意

有一块长 p 宽 q 高 $r+1$ 的切糕，在第 x 行第 y 列高度为 z 的位置切开 $(1 \le z \le r)$ 需要 $v(x,y,z)$ 的代价。

现在要求每个纵轴（即确定了 x 和 y）上要切开一个位置 $f(x,y)$ （$1 \le f(x,y) \le r$），并且相邻纵轴（每条纵轴与前后左右至多四条纵轴相邻）的切割高度差不超过 d。

要求最小化代价和，即

$$\min_f \sum_{x=1}^{p} \sum_{y=1}^{q} v(x,y,f(x,y))$$

$p,q,r \leqslant 40$。

解法

- 先不考虑相邻纵轴的高度限制，考虑如何把每种合法方案看成一个割。

解法

- 先不考虑相邻纵轴的高度限制，考虑如何把每种合法方案看成一个割。
- 只需要根据题意，把“切糕”的每一“小块”建一个点。在高度为 z 的位置切开实际上就是在点 $(x, y, z-1)$ 和点 (x, y, z) 之间切开。
于是从点 $(x, y, z-1)$ 到点 (x, y, z) 之间连一条边权为 $v(x, y, z)$ 的边。

解法

- 先不考虑相邻纵轴的高度限制，考虑如何把每种合法方案看成一个割。
- 只需要根据题意，把“切糕”的每一“小块”建一个点。在高度为 z 的位置切开实际上就是在点 $(x, y, z-1)$ 和点 (x, y, z) 之间切开。
  于是从点 $(x, y, z-1)$ 到点 (x, y, z) 之间连一条边权为 $v(x, y, z)$ 的边。
- 再考虑如果 $(x, y), (x', y')$ 两个纵列相邻，如何满足 $|f(x, y) - f(x', y')| \leq d$。出于对称性，我们只需要让 $f(x, y) \leq f(x', y') + d$ 即可，可以用同样的方法使得 $f(x', y') \leq f(x, y) + d$，从而满足限制。

解法

- 考虑到如果使 $f(x, y) = z_0$，割开 $(x, y, z_0 - 1) \to (x, y, z_0)$ 的边，就会使所有的 $(x, y, z) \quad (z < z_0)$ 都划分到 S 集合；所有 $(x, y, z) \quad (z \geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x, y) > f(x', y') + d$，那么 $(x', y', f(x', y'))$ 会属于 T 集合，而 $(x, y, f(x', y') + d)$ 会属于 S 集合。

解法

- 考虑到如果使 $f(x, y) = z_0$，割开 $(x, y, z_0 - 1) \to (x, y, z_0)$ 的边，就会使所有的 $(x, y, z) \quad (z < z_0)$ 都划分到 S 集合；所有 $(x, y, z) \quad (z \geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x, y) > f(x', y') + d$，那么 $(x', y', f(x', y'))$ 会属于 T 集合，而 $(x, y, f(x', y') + d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x, y, z + d)$ 向 (x', y', z) 连一条边权为 ∞ 的边即可。这样，由于 $(x', y', f(x', y'))$ 一定属于 T 集合，从而 $(x, y, f(x', y') + d)$ 也一定属于 T 集合，从而 $f(x, y) \leq f(x', y') + d$。

解法

- 考虑到如果使 $f(x, y) = z_0$，割开 $(x, y, z_0 - 1) \to (x, y, z_0)$ 的边，就会使所有的 $(x, y, z) \quad (z < z_0)$ 都划分到 S 集合；所有 $(x, y, z) \quad (z \geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x, y) > f(x', y') + d$，那么 $(x', y', f(x', y'))$ 会属于 T 集合，而 $(x, y, f(x', y') + d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x, y, z + d)$ 向 (x', y', z) 连一条边权为 ∞ 的边即可。这样，由于 $(x', y', f(x', y'))$ 一定属于 T 集合，从而 $(x, y, f(x', y') + d)$ 也一定属于 T 集合，从而 $f(x, y) \leq f(x', y') + d$。
- 综上，我们要连的边共有

解法

- 考虑到如果使 $f(x,y)=z_0$，割开 $(x,y,z_0-1)\to(x,y,z_0)$ 的边，就会使所有的 $(x,y,z)\quad(z<z_0)$ 都划分到 S 集合；所有 $(x,y,z)\quad(z\geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x,y)>f(x',y')+d$，那么 $(x',y',f(x',y'))$ 会属于 T 集合，而 $(x,y,f(x',y')+d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x,y,z+d)$ 向 (x',y',z) 连一条边权为 ∞ 的边即可。这样，由于 $(x',y',f(x',y'))$ 一定属于 T 集合，从而 $(x,y,f(x',y')+d)$ 也一定属于 T 集合，从而 $f(x,y)\leq f(x',y')+d$。
- 综上，我们要连的边共有
  1. 从 S 到每个 $(x,y,0)$，边权为 ∞；

解法

- 考虑到如果使 $f(x, y) = z_0$，割开 $(x, y, z_0 - 1) \to (x, y, z_0)$ 的边，就会使所有的 $(x, y, z) \quad (z < z_0)$ 都划分到 S 集合；所有 $(x, y, z) \quad (z \geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x, y) > f(x', y') + d$，那么 $(x', y', f(x', y'))$ 会属于 T 集合，而 $(x, y, f(x', y') + d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x, y, z + d)$ 向 (x', y', z) 连一条边权为 ∞ 的边即可。这样，由于 $(x', y', f(x', y'))$ 一定属于 T 集合，从而 $(x, y, f(x', y') + d)$ 也一定属于 T 集合，从而 $f(x, y) \leq f(x', y') + d$。
- 综上，我们要连的边共有
    1. 从 S 到每个 $(x, y, 0)$，边权为 ∞；
    2. 从每个 (x, y, r) 到 T，边权为 ∞；

解法

- 考虑到如果使 $f(x, y) = z_0$，割开 $(x, y, z_0 - 1) \to (x, y, z_0)$ 的边，就会使所有的 $(x, y, z) \quad (z < z_0)$ 都划分到 S 集合；所有 $(x, y, z) \quad (z \geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x, y) > f(x', y') + d$，那么 $(x', y', f(x', y'))$ 会属于 T 集合，而 $(x, y, f(x', y') + d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x, y, z + d)$ 向 (x', y', z) 连一条边权为 ∞ 的边即可。这样，由于 $(x', y', f(x', y'))$ 一定属于 T 集合，从而 $(x, y, f(x', y') + d)$ 也一定属于 T 集合，从而 $f(x, y) \leq f(x', y') + d$。
- 综上，我们要连的边共有
  1. 从 S 到每个 $(x, y, 0)$，边权为 ∞；
  2. 从每个 (x, y, r) 到 T，边权为 ∞；
  3. 从 $(x, y, z - 1)$ 到 (x, y, z)，边权为 $v(x, y, z)$；

解法

- 考虑到如果使 $f(x,y)=z_0$，割开 $(x,y,z_0-1)\to(x,y,z_0)$ 的边，就会使所有的 $(x,y,z)\quad(z<z_0)$ 都划分到 S 集合；所有 $(x,y,z)\quad(z\geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x,y)>f(x',y')+d$，那么 $(x',y',f(x',y'))$ 会属于 T 集合，而 $(x,y,f(x',y')+d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x,y,z+d)$ 向 (x',y',z) 连一条边权为 ∞ 的边即可。这样，由于 $(x',y',f(x',y'))$ 一定属于 T 集合，从而 $(x,y,f(x',y')+d)$ 也一定属于 T 集合，从而 $f(x,y)\leq f(x',y')+d$。
- 综上，我们要连的边共有
  1. 从 S 到每个 $(x,y,0)$，边权为 ∞；
  2. 从每个 (x,y,r) 到 T，边权为 ∞；
  3. 从 $(x,y,z-1)$ 到 (x,y,z)，边权为 $v(x,y,z)$；
  4. 从 (x,y,z) 到 $(x',y',z-d)$，边权为 ∞，其中 $(x,y),(x',y')$ 相邻，$d\leq z\leq r$。

解法

- 考虑到如果使 $f(x, y) = z_0$，割开 $(x, y, z_0 - 1) \to (x, y, z_0)$ 的边，就会使所有的 $(x, y, z) \quad (z < z_0)$ 都划分到 S 集合；所有 $(x, y, z) \quad (z \geq z_0)$ 划分到 T 集合。如果条件不成立，即 $f(x, y) > f(x', y') + d$，那么 $(x', y', f(x', y'))$ 会属于 T 集合，而 $(x, y, f(x', y') + d)$ 会属于 S 集合。
- 为避免这种状况发生，只需要从每个 $(x, y, z + d)$ 向 (x', y', z) 连一条边权为 ∞ 的边即可。这样，由于 $(x', y', f(x', y'))$ 一定属于 T 集合，从而 $(x, y, f(x', y') + d)$ 也一定属于 T 集合，从而 $f(x, y) \leq f(x', y') + d$。
- 综上，我们要连的边共有
  1. 从 S 到每个 $(x, y, 0)$，边权为 ∞；
  2. 从每个 (x, y, r) 到 T，边权为 ∞；
  3. 从 $(x, y, z - 1)$ 到 (x, y, z)，边权为 $v(x, y, z)$；
  4. 从 (x, y, z) 到 $(x', y', z - d)$，边权为 ∞，其中 $(x, y), (x', y')$ 相邻，$d \leq z \leq r$。
- 求出最小割即为答案。(PS：事实上可以把所有的 $(x, y, 0)$ 和 S 合并成一个点，(x, y, r) 和 T 合并成一个点)

例子

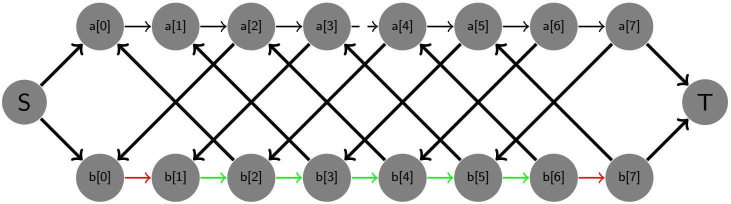

Figure: 两个相邻列之间的连边，加粗边的边权均为 ∞。由于列内的边权对于例子来说并不重要，这里并不标注。图中 $r = 7, d = 2$，如果割开 a_3 和 a_4 之间的边，那么 b 列只有绿色边能割。
实际上，$b[0], b[1]$ 必定划分到 S 集合，而 $b[6], b[7]$ 必定划分到 T 集合，所以只能在 $b[1]$ 到 $b[6]$ 之间割掉一条边。

扩展

- 给定一个差分约束系统，第 i 个变量取 x_i 的代价是 f_{i,x_i}，并且所有变量取 $[0,d]$ 的整数值，求可行解的最小代价。
  变量数为 n。限制数为 m。

扩展

- 给定一个差分约束系统，第 i 个变量取 x_i 的代价是 f_{i,x_i}，并且所有变量取 $[0,d]$ 的整数值，求可行解的最小代价。
  变量数为 n。限制数为 m。
- 利用本题的思路，很容易对上述问题进行建模并形成一个 $O(nd)$ 个点，$O((n+m)d)$ 条边的最小割模型。

模型：费用流

- 给定一张有向图，其中有两个特定的点 S 和 T，每条边有两个权值，即容量和**费用**。

模型：费用流

- 给定一张有向图，其中有两个特定的点 S 和 T，每条边有两个权值，即容量和**费用**。
- 有向边的费用代表了单位流量的运送价格（实际上不一定非负）。

模型：费用流

- 给定一张有向图，其中有两个特定的点 S 和 T，每条边有两个权值，即容量和**费用**。
- 有向边的费用代表了单位流量的运送价格（实际上不一定非负）。
- 求最大化流量的情况下的最小费用和。

算法

- 每次按费用找到 S 到 T 的最短路径并增广即可。可以证明只要初始图无负环，这个算法就是正确的。

算法

- 每次按费用找到 S 到 T 的最短路径并增广即可。可以证明只要初始图无负环，这个算法就是正确的。
- 有时候不要求流量最大而只要求费用最小，这时候可以在增广到单位费用 >0 时就停下来（因为增广过程中的单位费用，即 S 到 T 的最短路，是单调不降的）。

题意

有一张 n 个点 m 条边的带权 DAG（点和边都有权值）。
要求用若干条不相交的路径覆盖所有点（即每个点属于恰好一条路径）；
最小化路径上的边权与所有路径的起点的点权之和。
$n \leqslant 800, m \leqslant 15000$。

解法

- 不带权的路径覆盖问题可以转化成最大流来解决，于是我们考虑在不带权的路径覆盖的图上增加费用。

解法

- 不带权的路径覆盖问题可以转化成最大流来解决，于是我们考虑在不带权的路径覆盖的图上增加费用。
- 不带权的时候，我们把每个点 u 拆成 u 和 u'，$S \to u, u' \to T$ 都有一条容量为 1 的边；
  原图中每有一条边 $u \to v$，就连一条 $u \to v'$ 的容量为 1 的边。
  最大流的时候如果 $u \to v'$ 的边满流了，那么说明路径覆盖中我们要选择 $u \to v$ 的边。

解法

- 不带权的路径覆盖问题可以转化成最大流来解决，于是我们考虑在不带权的路径覆盖的图上增加费用。
- 不带权的时候，我们把每个点 u 拆成 u 和 u'，$S \to u, u' \to T$ 都有一条容量为 1 的边；
  原图中每有一条边 $u \to v$，就连一条 $u \to v'$ 的容量为 1 的边。
  最大流的时候如果 $u \to v'$ 的边满流了，那么说明路径覆盖中我们要选择 $u \to v$ 的边。
- 很容易想到可以直接把边权作为 $u \to v'$ 的边的费用。那么点权怎么办?

解法

- 不带权的路径覆盖问题可以转化成最大流来解决，于是我们考虑在不带权的路径覆盖的图上增加费用。
- 不带权的时候，我们把每个点 u 拆成 u 和 u'，$S \to u, u' \to T$ 都有一条容量为 1 的边；
  原图中每有一条边 $u \to v$，就连一条 $u \to v'$ 的容量为 1 的边。
  最大流的时候如果 $u \to v'$ 的边满流了，那么说明路径覆盖中我们要选择 $u \to v$ 的边。
- 很容易想到可以直接把边权作为 $u \to v'$ 的边的费用。那么点权怎么办?
- 考虑到如果 u 作为某条路径的起点，那么 $u' \to T$ 的边没有流量。
  我们从 S 到 u' 连一条边，容量为 1，费用为 u 的点权，这样在最大流的要求下，$S \to u' \to T$ 就会流过 1 的流量，从而造成 u 的点权的费用。

解法

- 不带权的路径覆盖问题可以转化成最大流来解决，于是我们考虑在不带权的路径覆盖的图上增加费用。
- 不带权的时候，我们把每个点 u 拆成 u 和 u'，$S \to u, u' \to T$ 都有一条容量为 1 的边；
  原图中每有一条边 $u \to v$，就连一条 $u \to v'$ 的容量为 1 的边。
  最大流的时候如果 $u \to v'$ 的边满流了，那么说明路径覆盖中我们要选择 $u \to v$ 的边。
- 很容易想到可以直接把边权作为 $u \to v'$ 的边的费用。那么点权怎么办?
- 考虑到如果 u 作为某条路径的起点，那么 $u' \to T$ 的边没有流量。
  我们从 S 到 u' 连一条边，容量为 1，费用为 u 的点权，这样在最大流的要求下，$S \to u' \to T$ 就会流过 1 的流量，从而造成 u 的点权的费用。
- 之后直接求最小费用最大流即可。

解法

- 另一种做法是考虑把所有的 $u \to v'$ 的边的费用减掉 v 的点权（或者把 $v' \to T$ 的费用设为 v 的点权的相反数），最后答案再加上所有点的点权和。

解法

- 另一种做法是考虑把所有的 $u \to v'$ 的边的费用减掉 v 的点权（或者把 $v' \to T$ 的费用设为 v 的点权的相反数），最后答案再加上所有点的点权和。
- 这样的话，如果 v 不是起点，它的点权就会在算费用的时候被减一次，从而最后不会计算。

解法

- 另一种做法是考虑把所有的 $u \to v'$ 的边的费用减掉 v 的点权（或者把 $v' \to T$ 的费用设为 v 的点权的相反数），最后答案再加上所有点的点权和。
- 这样的话，如果 v 不是起点，它的点权就会在算费用的时候被减一次，从而最后不会计算。
- 不过这样的话就要求最小费用可行流（即不要求最大化流量），而不是最大流。

解法

- 另一种做法是考虑把所有的 $u \to v'$ 的边的费用减掉 v 的点权（或者把 $v' \to T$ 的费用设为 v 的点权的相反数），最后答案再加上所有点的点权和。
- 这样的话，如果 v 不是起点，它的点权就会在算费用的时候被减一次，从而最后不会计算。
- 不过这样的话就要求最小费用可行流（即不要求最大化流量），而不是最大流。
- 事实上这样仿佛会比上一种方式快，大概因为边少并且增广次数少吧。

题意

要进行 n 天工作，第 i 天至少需要 A_i 人。

有 M 种人可以招募，第 i 种人从第 S_i 天工作到第 T_i 天，每个人的价格是 C_i 元。

要求在满足工作需求的前提下，最小化招募的花费。保证有解。

$n \leqslant 1000, m \leqslant 10000$。

解法

- 我们先来口胡一波。

解法

- 我们先来口胡一波。
- 对每一天建一个点，包括第 $n+1$ 天。
  从第 i 天的点向第 $i+1$ 天的点连一条容量为 ∞，费用为 0 的边，表示第 i 天的志愿者可以留到第 $i+1$ 天。

解法

- 我们先来口胡一波。
- 对每一天建一个点，包括第 $n+1$ 天。
  从第 i 天的点向第 $i+1$ 天的点连一条容量为 ∞，费用为 0 的边，表示第 i 天的志愿者可以留到第 $i+1$ 天。
- 如果有一种志愿者从第 u 天工作到第 v 天，每人价格是 C_i，就从 $v+1$ 到 u 连一条容量 ∞ 费用 C_i 的边，
  表示如果招募了一个这种志愿者，就会使第 u 天的人数 $+=1$，第 $v+1$ 天的人数 $-=1$。

解法

- 我们先来口胡一波。
- 对每一天建一个点，包括第 $n+1$ 天。
  从第 i 天的点向第 $i+1$ 天的点连一条容量为 ∞，费用为 0 的边，表示第 i 天的志愿者可以留到第 $i+1$ 天。
- 如果有一种志愿者从第 u 天工作到第 v 天，每人价格是 C_i，就从 $v+1$ 到 u 连一条容量 ∞ 费用 C_i 的边，
  表示如果招募了一个这种志愿者，就会使第 u 天的人数 $+=1$，第 $v+1$ 天的人数 $-=1$。
- 注意，每天的人是会留到下一天的，所以这就相当于把第 u 天到第 v 天的人数都加上了 1。

解法

- 接下来的问题是，如何让第 i 天的人数（实际上相当于 $i \to i+1$ 这条边的流量）不低于 A_i?

解法

- 接下来的问题是，如何让第 i 天的人数（实际上相当于 $i \to i+1$ 这条边的流量）不低于 A_i?
- 既然要求第 i 天的人数不小于 A_i，那就从 i “强行抽调”A_i 个人走，即从 i 到 T 连一条容量为 A_i 费用为 0 的边。

解法

- 接下来的问题是，如何让第 i 天的人数（实际上相当于 $i \to i+1$ 这条边的流量）不低于 A_i?
- 既然要求第 i 天的人数不小于 A_i，那就从 i “强行抽调”A_i 个人走，即从 i 到 T 连一条容量为 A_i 费用为 0 的边。
- 但是这样就会导致这 A_i 个人不见了，从而第 $i+1$ 天的时候我们白白损失了 A_i 个人。
  那我们再往第 $i+1$ 天强行塞进去 A_i 个人即可。

解法

- 接下来的问题是，如何让第 i 天的人数（实际上相当于 $i \to i+1$ 这条边的流量）不低于 A_i?
- 既然要求第 i 天的人数不小于 A_i，那就从 i “强行抽调”A_i 个人走，即从 i 到 T 连一条容量为 A_i 费用为 0 的边。
- 但是这样就会导致这 A_i 个人不见了，从而第 $i+1$ 天的时候我们白白损失了 A_i 个人。
  那我们再往第 $i+1$ 天强行塞进去 A_i 个人即可。
- 这样的话，由于从 S 出发的边的容量和等于进入 T 的容量和，求最大流之后一定可以把所有 S 出发的和流向 T 的边都满流；从而达到我们的目的。

解法

- 接下来的问题是，如何让第 i 天的人数（实际上相当于 $i \to i+1$ 这条边的流量）不低于 A_i?
- 既然要求第 i 天的人数不小于 A_i，那就从 i“强行抽调”A_i 个人走，即从 i 到 T 连一条容量为 A_i 费用为 0 的边。
- 但是这样就会导致这 A_i 个人不见了，从而第 $i+1$ 天的时候我们白白损失了 A_i 个人。
  那我们再往第 $i+1$ 天强行塞进去 A_i 个人即可。
- 这样的话，由于从 S 出发的边的容量和等于进入 T 的容量和，求最大流之后一定可以把所有 S 出发的和流向 T 的边都满流；从而达到我们的目的。
- 这样，$i \to i+1$ 的边的流量实际上是第 i 天并没有干活的人数，而第 i 天干活的 A_i 个人则从 i 流向了 T，再由 S 补充给 $i+1$。

解法

- 整理一下我们要连的边：首先一共有 $n+3$ 个点，S, T，以及每天的点 (包括 $n+1$)。

解法

- 整理一下我们要连的边：首先一共有 $n+3$ 个点，S, T，以及每天的点(包括 $n+1$)。
  1. $S \to i$，容量 A_{i-1}，费用 0；

解法

- 整理一下我们要连的边：首先一共有 $n+3$ 个点，S, T，以及每天的点 (包括 $n+1$)。
  1. $S \to i$，容量 A_{i-1}，费用 0；
  2. $i \to T$，容量 A_i，费用 0；

解法

- 整理一下我们要连的边：首先一共有 $n+3$ 个点，S, T，以及每天的点 (包括 $n+1$)。
  1. $S \to i$，容量 A_{i-1}，费用 0；
  2. $i \to T$，容量 A_i，费用 0；
  3. $T_i + 1 \to S_i$，容量 ∞，费用 C_i。

解法

- 整理一下我们要连的边：首先一共有 $n+3$ 个点，S, T，以及每天的点(包括 $n+1$)。
  1. $S \to i$，容量 A_{i-1}，费用 0；
  2. $i \to T$，容量 A_i，费用 0；
  3. $T_i + 1 \to S_i$，容量 ∞，费用 C_i。
- (注意上文的 S_i 和 T_i 是指志愿者工作开始和结束的时间，不要和 S, T 混淆)。
  一个小优化是，既然 S 到 i 连了容量为 A_{i-1} 的边，而 i 到 T 连了容量为 A_i 的边，那么可以去掉其中容量小的一条边。
  也就是说如果 $A_{i-1} > A_i$，就从 S 向 i 连容量为 $A_{i-1} - A_i$ 的边，否则则从 i 向 T 连容量为 $A_i - A_{i-1}$ 的边。

解法

- 整理一下我们要连的边：首先一共有 $n+3$ 个点，S, T，以及每天的点（包括 $n+1$）。
  1. $S \to i$，容量 A_{i-1}，费用 0；
  2. $i \to T$，容量 A_i，费用 0；
  3. $T_i + 1 \to S_i$，容量 ∞，费用 C_i。
- （注意上文的 S_i 和 T_i 是指志愿者工作开始和结束的时间，不要和 S, T 混淆）。
  一个小优化是，既然 S 到 i 连了容量为 A_{i-1} 的边，而 i 到 T 连了容量为 A_i 的边，那么可以去掉其中容量小的一条边。
  也就是说如果 $A_{i-1} > A_i$，就从 S 向 i 连容量为 $A_{i-1} - A_i$ 的边，否则则从 i 向 T 连容量为 $A_i - A_{i-1}$ 的边。
- 之后就只需要求出最小费用最大流即可。

拓展

- 仔细观察刚才的建图方式，我们会发现，实际上这相当于给了从 i 流向 $i+1$ 的流量的一个“下界”A_i、求出“最小费用环流”。

拓展

- 仔细观察刚才的建图方式，我们会发现，实际上这相当于给了从 i 流向 $i+1$ 的流量的一个“下界”A_i、求出“最小费用环流”。
- 所谓“环流”就是没有 S 与 T，每个点都要满足流量平衡。在没有下界限制的情况下这实际上是平凡的，因为零流就是合法的环流。
  而零流一般总是最小费用，除非图中有负环，这种情况不考虑。

拓展

- 仔细观察刚才的建图方式，我们会发现，实际上这相当于给了从 i 流向 $i+1$ 的流量的一个“下界”A_i、求出“最小费用环流”。
- 所谓“环流”就是没有 S 与 T，每个点都要满足流量平衡。在没有下界限制的情况下这实际上是平凡的，因为零流就是合法的环流。
  而零流一般总是最小费用，除非图中有负环，这种情况不考虑。
- 我们可以利用类似的方法解一个任意的给定边的上下界的问题，即如果某条边 $u \to v$ 的上界（容量）是 b，下界是 a，就把它的容量设为 $b-a$，而从 S 向 v、从 u 向 T 分别连一条容量为 a 的边，之后再以上面的方法合并每个点向 S、T 连的边即可。

拓展

- 仔细观察刚才的建图方式，我们会发现，实际上这相当于给了从 i 流向 $i+1$ 的流量的一个“下界”A_i、求出“最小费用环流”。
- 所谓“环流”就是没有 S 与 T，每个点都要满足流量平衡。在没有下界限制的情况下这实际上是平凡的，因为零流就是合法的环流。
而零流一般总是最小费用，除非图中有负环，这种情况不考虑。
- 我们可以利用类似的方法解一个任意的给定边的上下界的问题，即如果某条边 $u \to v$ 的上界（容量）是 b，下界是 a，就把它的容量设为 $b-a$，而从 S 向 v、从 u 向 T 分别连一条容量为 a 的边，之后再以上面的方法合并每个点向 S、T 连的边即可。
- 至于普通的上下界网络流（而并非“环流”），可以考虑连一条 $T \to S$ 的容量为 ∞ 的边，这样流量从 T 进入再从 S 出来，相当于环流。
注意这时候我们还是要新建两个点作为“超级源点”和“超级汇点”，而不能利用之前的 S 和 T。

拓展

- 仔细观察刚才的建图方式，我们会发现，实际上这相当于给了从 i 流向 $i+1$ 的流量的一个“下界”A_i、求出“最小费用环流”。
- 所谓“环流”就是没有 S 与 T，每个点都要满足流量平衡。在没有下界限制的情况下这实际上是平凡的，因为零流就是合法的环流。
  而零流一般总是最小费用，除非图中有负环，这种情况不考虑。
- 我们可以利用类似的方法解一个任意的给定边的上下界的问题，即如果某条边 $u \to v$ 的上界（容量）是 b，下界是 a，就把它的容量设为 $b-a$，而从 S 向 v、从 u 向 T 分别连一条容量为 a 的边，之后再以上面的方法合并每个点向 S、T 连的边即可。
- 至于普通的上下界网络流（而并非“环流”），可以考虑连一条 $T \to S$ 的容量为 ∞ 的边，这样流量从 T 进入再从 S 出来，相当于环流。
  注意这时候我们还是要新建两个点作为“超级源点”和“超级汇点”，而不能利用之前的 S 和 T。
- 这方面的知识，相信大家搞懂这道题的做法之后很容易理解，限于篇幅和讲课时间，这里不再赘述。

网络流算法作为一个常考的知识点，模型有很多种。

这里挑选了若干常见模型来讲解，而实际上的类型要远远多于此。

希望这能够启发大家理解各种网络流模型的构建想法，而不是单纯地了解这几道题目的做法。

相信这个课件，可以给拼搏于 OI 的逐梦之路上的你，提供一个有力的援助。

谢谢大家！