提高算法班

分层图最短路、同余最短路、差分约束

Mas

#2242、飞行路线

题目描述

$Alice$ 和 Bob 现在要乘飞机旅行,他们选择了一家相对便宜的航空公司

该航空公司一共在 n 个城市设有业务,设这些城市分别标记为 0 到 $n-1$

一共有 m 种航线,每种航线连接两个城市,并且航线有一定的价格

$Alice$ 和 Bob 现在要从一个城市沿着航线到达另一个城市,途中可以进行转机

航空公司对他们这次旅行也推出优惠,他们可以免费在最多 k 种航线上搭乘飞机

那么 $Alice$ 和 Bob 这次出行最少花费多少?

输入格式

第一行三个整数 n, m, k ,分别表示城市数,航线数和免费乘坐次数

接下来一行两个整数 s, t ,分别表示他们出行的起点城市编号和终点城市编号

接下来 m 行,每行三个整数 a, b, c
表示存在一种航线,能从城市 a 到达城市 b ,或从城市 b 到达城市 a ,价格为 c

输出格式

输出一行一个整数,为最少花费

数据规模

对于 30% 的数据, $2 \le n \le 50, 1 \le m \le 300, k = 0$

对于 50% 的数据, $2 \le n \le 600, 1 \le m \le 6 \times 10^3, 0 \le k \le 1$

对于 100% 的数据, $2 \le n \le 10^4, 1 \le m \le 5 \times 10^4, 0 \le k \le 10$

#2242、飞行路线

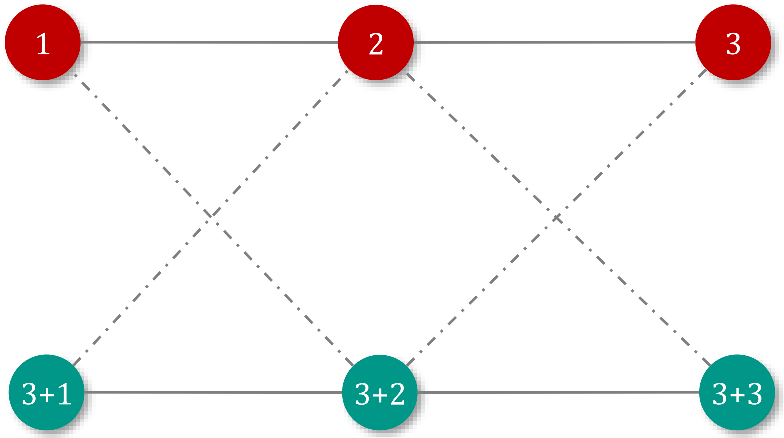
思路1

可以将图复制 k 份，第 i 层图所有点的编号为 $i\times n+1\sim i\times n+n$

对于每一层内, 按照原图关系在新图中连接边

在相邻的两份 $i,i+1$ 层图

- $i\times n+u\leftrightarrow(i+1)\times n+v$ 连一条权值为 0 的边
- $i\times n+v\leftrightarrow(i+1)\times n+u$ 连一条权值为 0 的边



这样图被分层了 k 层，每层之间有一些边权为 0 的边

在求最短路时，若经过层与层之间边权为 0 的边时，意味着使用了一次免费的机会

答案为 $\min_{i=0}^{k}\{\mathrm{dis}_{i\times n+et}\}$

若使用 Dijkstra 算法求解， 时间复杂度 $O(k\,(n+m)\log(km))$

#2242、飞行路线

思路2

令 $\mathrm{dis}_{v,j}$ 表示 从起点到达 v 使用 j 次免费机会的最短路径长度

记当前点为 v，u 为 v 的前驱节点，w 为当前边权

到达点 v 时可使用一次免费机会即 $\mathrm{dis}_{u,j-1}$

到达点 v 时可不使用免费机会即 $\mathrm{dis}_{u,j}+w$

$$\mathrm{dis}_{v,j}=\min\{\ \min\{\ \mathrm{dis}_{u,j-1}\ \},\min\{\ \mathrm{dis}_{u,j}\ +\ w\ \}\ \}$$

当 $j\ -1\ \geq k$ 时 $\mathrm{dis}_{v,j}=\infty$

在 Dijkstra 过程中转移，对于每一次状态更新成功，需让节点重新入队以更新后续状态

答案为 $\min_{i=0}^{k}\{\mathrm{dis}_{et,i}\}$

上述过程可看作将各个点拆分成 $k+1$ 份, 每份代表使用了对应次数的免费机会

时间复杂度 $O(k\ (n+m)\log{(km)}\ )$

#2744、Mas的电梯

题目描述

Mas 的家是一幢 h 层的摩天大楼

由于客人越来越多, Mas 改造了一个电梯,使得访客可以更方便的上楼

经过改造, Mas 的电梯可以采用以下四种方式移动:

- 向上移动 x 层
- 向上移动 y 层
- 向上移动 z 层
- 回到第一层

现在 Mas 想知道,他可以乘坐电梯前往的楼层数

输入格式

第一行一个整数 h ,表示摩天大楼的层数

第二行三个正整数,分别表示题目中的 x, y, z

输出格式

一行一个整数,表示可以到达的楼层数

对于前 40% 的数据，背包 DP 计数，时间复杂度$O(h)$

考虑仅由 y, z 凑出的楼层数为 num，即 $\text{num} = ay + bz$

剩下的全部楼层由 x 凑出的方案为 $\left\lfloor \frac{h-\text{num}}{x} \right\rfloor + 1$

只需要找出所有的 num 即可得到答案

对于 $\text{num}_1 = \text{num}_2 + kx$ 即 $\text{num}_1 \equiv \text{num}_2 \pmod x$ 时

num_1与 num_2 会重复贡献

数据规模

对于 10% 的数据 $1 \le h \le 50, 1 \le x, y, z \le 50$

对于 40% 的数据 $1 \le h \le 100000, 1 \le x, y, z \le 100$

对于全部的数据 $1 \le h \le 2^{63} - 1, 1 \le x, y, z \le 10^5$

#2744、Mas的电梯

不妨假设 $x \le y \le z$

令 d_i 为只通过 $+y$ 或 $+z$ 得到满足 $d_i \bmod x = i$ 能够达到的最低楼层

即 $+y$ 或 $+z$ 操作后能得到的模 x 下与 i 同余的最小非负整数(从这一层开始每向上 x 层都能到)

将 $0 \sim x-1$ 的范围内的所有点 i

- $i \longrightarrow (i+y)\ \%\ x$ 建立边权为 y 的有向边
- $i \longrightarrow (i+z)\ \%\ x$ 建立边权为 z 的有向边

因大楼最低为 1 层 , 令 $d_1 = 1$ 跑一遍最短路即可求出所有 dis_i

答案为

$$\sum_{i=0}^{x-1}\left(\left\lfloor\frac{h-d_i}{x}\right\rfloor+1\right)$$

时间复杂度取决于最短路算法

Bellman-Ford 的其他优化

除 SPFA 外, Bellman – Ford 还有其他形式的优化

这些优化在部分图上效果明显，但在某些特殊图上，最坏复杂度可能达到**指数级**

堆优化

将队列换成堆，与 Dijkstra 的区别是允许一个点多次出队松弛

在有负权边的图可能被卡成指数级复杂度

栈优化

将队列换成栈(即将原来的 BFS 过程变成 DFS)

在寻找负环时可能具有更高效率，但最坏时间复杂度仍然为指数级

Bellman-Ford 的其他优化

LLL 优化

将普通队列换成双端队列

每次将入队结点距离和队内距离平均值比较，如果更大则插入至队尾，否则插入队首

SLF 优化

将普通队列换成双端队列

每次将入队结点距离和队首比较，如果更大则插入至队尾，否则插入队首

D´Esopo-Pape 算法

将普通队列换成双端队列

若一个节点之前没有入队将其插入队尾，否则插入队首

更多优化以及针对这些优化的 Hack 方法, 可参考 fstqwq 在知乎上的回答

负环

边权和为负数的环

当存在负环时,起点到终点可能不存在最短路

Floyd

执行完循环后, 若存在 $\text{dis}_{i,i}$ 的值是负数,说明存在负环

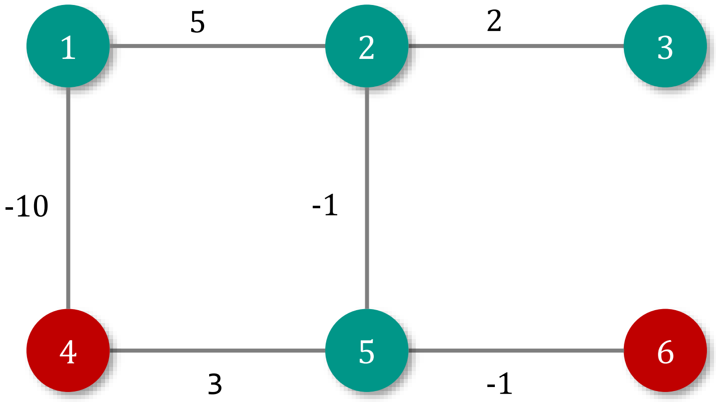
时间复杂度 $O(n^3)$

SPFA

若 s 到 e 之间存在 负环 , 每转过一次负环 , 最短路长度会减小(无限减小)

负环上的点 , 会入队超过 $n-1$ 次

✓ 负环上的点最短路径经过的边数会超过 $n-1$



#681、虫洞 Wormholes

题目描述

FJ 在他的农场中闲逛时发现了许多虫洞

虫洞可以看作一条十分奇特的有向边,并可以使你返回到过去的一个时刻(相对你进入虫洞之前)

FJ 的每个农场有 M 条小路(无向边)连接着 N (从 1 到 N 标号)块地,并有 W 个虫洞

现在 FJ 想借助这些虫洞来回到过去(***在出发时刻之前回到出发点***),请你告诉他能办到吗。 FJ 将向你提供 F 个农场的地图

没有小路会耗费你超过 10^4 秒的时间,当然也没有虫洞回帮你回到超过 10^4 秒以前

输入格式

第一行一个整数 F ,表示农场个数

对于每个农场:第一行三个整数 N, M, W

接下来 M 行,每行三个数 S, E, T ,表示在标号为 S 的地与标号为 E 的地中间有一条用时 T 秒的小路

接下来 W 行,每行三个数 S, E, T ,表示在标号为 S 的地与标号为 E 的地中间有一条可以使 FJ 到达 T 秒前的虫洞

输出格式

输出共 F 行

如果 FJ 能在第 i 个农场实现他的目标,就在第 i 行输出 `YES` ,否则输出 `NO`

数据范围

对于全部数据, $1 \leq F \leq 5, 1 \leq N \leq 500, 1 \leq M \leq 2500, 1 \leq W \leq 200, 1 \leq S, E \leq N, |T| \leq 10^4$

判断图上是否存在负环

图上可能存在多个连通块

建立汇点向所有点连一条权值为 0 的边

也可在初始时将所有点都入队

#895、Sightseeing Cows

题目描述

给定一张 L 个点、 P 条边的有向图,每个点都有一个权值 f_i ,每条边都有一个权值 t_i

求图中的一个环,使**环上各点的权值之和**除以**环上各边的权值之和**最大

输出这个最大值

注意：数据保证至少存在一个环

输入格式

第一行包含两个整数 L 和 P

接下来 L 行每行一个整数,表示 f_i

再接下来 P 行,每行三个整数 a, b, t_i ,表示点 a 和 b 之间存在一条边,边的权值为 t_i

输出格式

输出一个数表示结果,保留两位小数

数据范围

对于全部的数据 $2 \le L \le 1000, 2 \le P \le 5000, 1 \le f_i, t_i \le 1000$

#895、Sightseeing Cows

要使得环上 $\frac{\sum f_i}{\sum t_i}$ 最大

若存在一个环满足 $\frac{\sum f_i}{\sum t_i} > x$

那么

$$\sum f_i > x\sum t_i \Rightarrow \sum t_i x - \sum f_i < 0$$

原图边 (u,v,w) 边权看作 $wx - f_u$ (将点权放到出边上)

问题转换为图上否存在负环

二分答案找出 x 上界即可

差分约束

差分约束系统 是一种特殊的 n 元一次不等式组，它包含 n 个变量 $x_1, x_2, \dots, x_n$ 及 m 个约束条件

每个约束条件由其中两个变量做差构成，形如 $x_i - x_j \le c_k$

其中 $1 \le i, j \le n, i \ne j, 1 \le k \le m$ 且 c_k 是常数

需要求一组解 $x_1 = a_1, x_2 = a_2, \dots, x_n = a_n$

使得所有的约束条件得到满足，或判断出无解

$$\begin{cases} x_{i_1} - x_{i_1'} \le c_1 \\ x_{i_2} - x_{i_2'} \le c_2 \\ \cdots \\ x_{i_m} - x_{i_m'} \le c_m \end{cases}$$

容易发现若 $\{a_1, a_2, \dots, a_n\}$ 是该差分约束系统的一组解

对于任意的常数 d

$\{a_1 + d, a_2 + d, \dots, a_n + d\}$ 显然也是该差分约束系统的一组解

差分约束

每个约束条件 $x_i - x_j \le c_k$ 都可以变形成 $x_i \le x_j + c_k$

这与单源最短路中的三角形不等式 $\text{dis}_v \le \text{dis}_u + w$ 非常相似

将每个变量 x_i 看做图中的一个结点

对于每个约束条件 $x_i - x_j \le c_k$ 从结点 j 向结点 i 连一条长度为 c_k 的有向边

设 $\text{dis}_0 = 0$ 并向每一个点连一条权重为 0 边,跑单源最短路

若图中存在负环，则给定的差分约束系统无解,否则 $x_i = \text{dis}_\text{i}$ 为该差分约束系统的一组解

且这组解满足性质：每个点都尽可能地取到了最大值

一般使用 $\text{Bellman} - \text{Ford}$ 或 SPFA 判断图中是否存在负环

最坏时间复杂度为 $O(nm)$

为什么存在负环就无解?

#1270、差分约束系统

题目描述

给出一组包含 m 个不等式,有 n 个未知数的形如:

$$\begin{cases} x_{c_1} - x_{c'_1} \le y_1 \\ x_{c_2} - x_{c'_2} \le y_2 \\ \cdots \\ x_{c_m} - x_{c'_m} \le y_m \end{cases}$$

的不等式组,求任意一组满足这个不等式组的解

输入格式

第一行为两个正整数 n, m ,代表未知数的数量和不等式的数量

接下来 m 行,每行包含三个整数 c, c', y ,代表一个不等式 $x_c - x_{c'} \le y$

输出格式

输出一行 n 个数,表示 $x_1, x_2 \cdots x_n$ 的一组可行解

如果有多组解,请输出 int 范围内任意一组,无解请输出 `NO`

数据范围

对于 100% 的数据, $1 \le n, m \le 5 \times 10^3, -10^4 \le y \le 10^4, 1 \le c, c' \le n, c \neq c'$

#1270、差分约束系统

从 c' 向 c 连一条权值为 w 的有向边

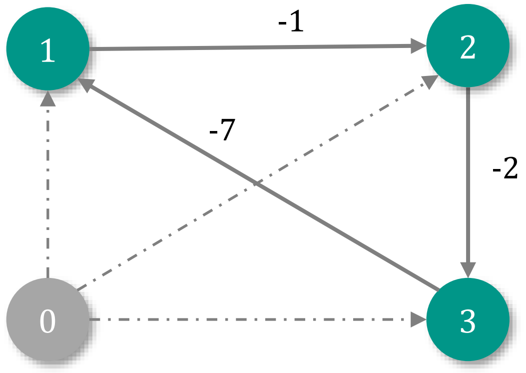
考虑到可能图不连通，建立一个超级点 0，从 0 向所有点连一条权值为 0 的边 $(x_i \le x_0)$

dis_0 初始值设为 0 (说明 $x_i \le 0$)

在 SPFA 中 求解最短路，dis 中的值 就为 x_i 的一个可行解

右图原不等式组为

$$\begin{cases} x_2 - x_1 \le -1 \\ x_3 - x_2 \le -2 \\ x_1 - x_3 \le -7 \end{cases}$$



三式子相加得到 $x_1 \le x_1 - 10$，当图上出现负环，说明无解

若要求满足所有约束且 $\forall 1 \le i \le n, x_i \le W$ 的最大值?

#861、糖果

题目描述

幼儿园里有 N 个小朋友,老师现在想要给这些小朋友们分配糖果
要求每个小朋友都要分到糖果但是小朋友们也有嫉妒心,总是会提出一些要求

比如小明不希望小红分到的糖果比他的多,于是在分配糖果的时候,老师需要满足小朋友们的 K 个要求

幼儿园的糖果总是有限的,老师想知道他至少需要准备多少个糖果,才能使得每个小朋友都能够分到糖果
并且满足小朋友们所有的要求

输入格式

输入的第一行是两个整数 N , K

接下来 K 行,表示这些点需要满足的关系,每行 3 个数字, X , A , B

- 如果 $X=1$, 表示第 A 个小朋友分到的糖果必须和第 B 个小朋友分到的糖果一样多
- 如果 $X=2$, 表示第 A 个小朋友分到的糖果必须少于第 B 个小朋友分到的糖果
- 如果 $X=3$, 表示第 A 个小朋友分到的糖果必须不少于第 B 个小朋友分到的糖果
- 如果 $X=4$, 表示第 A 个小朋友分到的糖果必须多于第 B 个小朋友分到的糖果
- 如果 $X=5$, 表示第 A 个小朋友分到的糖果必须不多于第 B 个小朋友分到的糖果;

输出格式

输出一行,表示老师至少需要准备的糖果数,如果不能满足小朋友们的所有要求,就输出 -1

数据规模

对于 30% 的数据,保证 $N \leq 100$
对于 100% 的数据,保证 $N \leq 100000$
对于所有的数据,保证 $K \leq 100000, 1 \leq X \leq 5, 1 \leq A, B \leq N$

#861、糖果

- $X = 1$

  $a = b$ 可以拆分为 $a \ge b$ 和 $a \le b$

- $X = 2$

  $a < b$ 可以转化为 $b \ge a + 1$

- $X = 3$

  $a \ge b$ 不需要转换

- $X = 4$

  $a > b$可以转换为 $a \ge b + 1$

- $X = 5$

  $a \le b$ 不需要转换

要求满足所有约束的最小值

在求解最长路时有 $\mathrm{dis}_v \ge \mathrm{dis}_u + w$

每个约束条件 $x_i - x_j \le c_k$ 都可以变形成 $x_j - x_i \ge -c_k$

#861、糖果

将每个变量 x_i 看做图中的一个结点

对于每个约束条件 $x_i - x_j \geq c_k$ 变形为 $x_i \geq x_j + c_k$

从结点 j 向结点 i 连一条长度为 c_k 的有向边，求解最长路即可

要求所有同学都能分到，且分到数量非负

通过建立超级点 0，从 0 到所有点建立有向边权值为 0 同时令 $\text{dis}_0 = 1$

本题需要需要使用 **栈优化** 的 Bellman – Ford 才能通过

求解最长路时不能使用 Dijkstra， 反例见 《12、经典最短路问题》P13

本题正解为 tarjan缩点 + 拓扑排序

#1723、全源最短路

题目描述

给定一个包含 n 个结点和 m 条带权边的有向图,求所有点对间的最短路径长度,一条路径的长度定义为这条路径上所有边的权值和

注意:边权**可能**为负,且图中**可能**存在重边和自环

输入格式

第 1 行: 2 个整数 n, m ,表示给定有向图的结点数量和有向边数量

接下来 m 行: 每行 3 个整数 u, v, w ,表示有一条权值为 w 的有向边从编号为 u 的结点连向编号为 v 的结点

输出格式

若图中存在负环,输出仅一行 -1

若图中不存在负环: 输出 n 行: 令 $dis_{i,j}$ 为从 i 到 j 的最短路,在第 i 行输出 $\sum_{j=1}^{n} j \times dis_{i,j}$

注意这个结果可能超过 int 存储范围

如果不存在从 i 到 j 的路径,则 $dis_{i,j} = 10^9$

如果 $i = j$,则 $dis_{i,j} = 0$

数据规模

对于 20% 的数据, $1 \le n \le 100$,不存在负环

对于另外 20% 的数据, $w \ge 0$

对于 100% 的数据, $1 \le n \le 3 \times 10^3$, $1 \le m \le 6 \times 10^3$, $1 \le u, v \le n$, $-3 \times 10^5 \le w \le 3 \times 10^5$

若图中存在负环时无解

Floyd 算法求解时间复杂度 $O(n^3)$

n 次 Bellman– Ford 算法求解时间复杂度 $O(n^2m)$

无法拿到全部的分数

若图中不存在负边权 n 次 Dijkstra 算法可通过

#1723、全源最短路

考虑将边权改造为非负

直接将边权加定值改造显然不可行，反例见 《12、经典最短路问题》P10

Johnson 全源最短路径算法

该算法可求出无负环图上任意两点间最短路径，由 Donald B. Johnson 在 1977 年提出

新建一个虚拟节点(不妨设其编号为 0)

从 0 向其他所有点连一条边权为 0 的边

接下来用 Bellman–Ford 算法求出从 0 号点到其他所有点的最短路，记为 h_i

假如存在一条从 u 点到 v 点,边权为 w 的边，则将该边的边权重设为 $w + h_u - h_v$

接下来以每个点为起点，跑 n 次 Dijkstra 算法即可求出任意两点间的最短路

#1723、全源最短路

实验舱 青少年编程 LAB 走近科学 走进名校

正确性

在重新标记后的图上，从 $s \to t$ 的一条路径 $s \to p_1 \to p_2 \to \cdots \to p_k \to t$ 的长度为：

$$w(s,p_1) + h_s - h_{p_1} + w(p_1,p_2) + h_{p_1} - h_{p_2} + \cdots + w(p_k,t) + h_{p_k} - h_t$$

整理得：

$$w(s,p_1) + w(p_1,p_2) + \cdots + w(p_k,t) + h_s - h_t$$

无论从 s 到 t 选择哪一条路径 $h_s - h_t$ 的值不变

为了方便，下面我们就把 h_i 称为 i 点的势能

新图中 $s \to t$ 的最短路的长度表达式由两部分组成，即原图中 $s \to t$ 的最短路 与 两点间的势能差

因为两点间势能的差为定值，因此原图上 $s \to t$ 的最短路与新图上 $s \to t$ 的最短路相对应

#1723、全源最短路

考虑 Bellman– Ford 算法中求解势能数组 h

根据三角形不等式，图上任意一边 (u, v) 上两点满足

$$h_v \leq h_u + w(u, v)$$

该边重新标记后的边权为

$$w'(u, v) = w(u, v) + h_u - h_v \geq 0$$

即新图上的边权均非负

一开始的 Bellman– Ford 算法并不是时间上的瓶颈

若使用 priority_queue 实现 Dijkstra 算法，该算法的时间复杂度是 $O(n\,(n + m)\log\, m)$

实验舱
青少年编程
走近科学 走进名校

谢谢观看