最小生成树，最短路径

尹雪淳

吉林大学

2025 年 1 月 17 日

目录

最小生成树

生成树是从一张无向连通图中选取一些边构成一张新图，使得这张图是是一棵树

最小生成树即是让上述的生成树的边权和最小
同时，最小生成树也会有一些性质

1. 在最小生成树上，两个点路径上经过的边权最小值即是这两个点在原图中所有路径中可能经过边的边权最小值
2. 对于原图中所有合法的最小生成树，他们的边权构成是相同的
3. 最小生成树是瓶颈生成树的充分不必要条件，即最小生成树一定是瓶颈生成树，而瓶颈生成树不一定是最小生成树
   瓶颈生成树为在原图的所有生成树中，最大的边权最小

我们接下来将介绍两种求最小生成树的方法，即 Kruskal 和 Prim

Kruskal

Kruskal 是一种贪心算法，即从未选边中选取一条边权最小的边，满足选取该边后，原图不出环

那么想要找到这条边，我们需要同时判断两个东西

1. 当前的最小边权的边
2. 这条边所连接的两个点的连通性

Kruskal

对于第一个问题，我们考虑可以用并查集来维护两个点之间的连通性，即如果两个点在并查集内部有共同祖先，那么即说明这两个点是联通的

对于第二个问题，我们可以对原图的边进行排序，然后按顺序判断是否满足条件 1 即可

Kruskal

对于正确性，我们考虑当前待加入的边 e

- 如果它属于最终的正确的生成树，那么显然正确
- 如果它不属于最终的正确的生成树，那么我们加入后一定会有一个环，考虑这个环上不是点 e 的任意一条边 e'
  - 如果 e' 的边权大于 e，那么我们构建最小生成树的时候应该会先加入 e 后加入 e'，那么含有 e 的生成树就是比正确的生成树更优的，这显然与事实不符
  - 如果 e' 的边权小于 e，那么我们在加入 e 之前肯定已经加入 e' 了。因为 e' 具有任意性，所以在想要加入 e 这条边的时候，这条边的两个端点已经联通，自然不会加入 e 这条边

Prim

如果说 Kruskal 是找满足条件的边，那么 Prim 就是找满足条件的点

即在目前没有加入连通块的点中，选取一个距离连通块最近的点加入连通块

那么想要找到这个点，我们需要思考三个东西

1. 现在有哪些没有加入连通块的点
2. 这些点距离连通块的距离是多少
3. 距离最近的点是什么

Prim

事实上，我们发现，我们其实只需要考虑和当前连通块有连边的那些点，因为其它点想要到达连通块一定需要经过那些有连边的点和那些边
也就是说，我们只需要保存当前和连通块有连边的点就行了
同时，我们也就是只需要在每次加入新的点之后，更新一下这个点连接的点和他们到连通块的距离即可

这样，我们只需要用一个堆来存储所有和连通块有边相连的点即可
然后每次取出距离连通块最近的点，然后向堆中加入其相邻的点，或者更新相邻点到当前连通块的距离

【一本通图最小生成树】最短网络 (agrinet)

农民约翰被选为他们镇的镇长！他其中一个竞选承诺就是在镇上建立起互联网，并连接到所有的农场。当然，他需要你的帮助。约翰已经给他的农场安排了一条高速的网络线路，他想把这条线路共享给其他农场。为了用最小的消费，他想铺设最短的光纤去连接所有的农场。你将得到一份各农场之间连接费用的列表，你必须找出能连接所有农场并所用光纤最短的方案。每两个农场间的距离不会超过 100000。
农场的个数，N（3 <= N <= 100）。

【一本通图最小生成树】最短网络 (agrinet)

这道题要求出总光纤长度最短的方案，即是最小生成树定义

首先，题目给的是邻接矩阵的形式，应当转化成 Kruskal 或 Prim 对应的图的存储形式

然后跑最小生成树，最后树的边权和即是答案

【一本通图最小生成树】局域网

某个局域网内有 n(n<=100) 台计算机，由于搭建局域网时工作人员的疏忽，现在局域网内的连接形成了回路，我们知道如果局域网形成回路那么数据将不停的在回路内传输，造成网络卡的现象。

因为连接计算机的网线本身不同，所以有一些连线不是很畅通，我们用 f(i,j) 表示 i,j 之间连接的畅通程度 (f(i,j)<=1000)，f(i,j) 值越小表示 i,j 之间连接越通畅，f(i,j) 为 0 表示 i,j 之间无网线连接。现在我们需要解决回路问题，我们将除去一些连线，使得网络中没有回路，并且被除去网线的 Σf(i,j) 最大，请求出这个最大值。

【一本通图最小生成树】局域网

题目中要求在保证联通的情况下使得去除的网线边权和最大，即保留的边权和最小，我们考虑求最小生成树

答案即是原图边权和减去最小生成树边权和

【一本通图最小生成树】繁忙的都市

城市 C 是一个非常繁忙的大都市，城市中的道路十分的拥挤，于是市长决定对其中的道路进行改造。
城市 C 的道路是这样分布的：城市中有 n 个交叉路口，有些交叉路口之间有道路相连，两个交叉路口之间最多有一条道路相连接。这些道路是双向的，且把所有的交叉路口直接或间接的连接起来了。每条道路都有一个分值，分值越小表示这个道路越繁忙，越需要进行改造。
但是市政府的资金有限，市长希望进行改造的道路越少越好，于是他提出下面的要求：

1. 改造的那些道路能够把所有的交叉路口直接或间接的连通起来。
2. 在满足要求 1 的情况下，改造的道路尽量少。
3. 在满足要求 1、2 的情况下，改造的那些道路中分值最大的道路分值尽量小。

任务：作为市规划局的你，应当作出最佳的决策，选择那些道路应当被修建。

【一本通图最小生成树】繁忙的都市

观察题目的要求，我们发现这是让我们求原图的瓶颈生成树
根据之前的性质，最小生成树就是瓶颈生成树

于是我们就求原图的最小生成树，在求的时候记录生成树的边，取最大值就是答案

【一本通图最小生成树】联络员 (liaison)

Tyvj 已经一岁了，网站也由最初的几个用户增加到了上万个用户，随着 Tyvj 网站的逐步壮大，管理员的数目也越来越多，现在你身为 Tyvj 管理层的联络员，希望你找到一些通信渠道，使得管理员两两都可以联络 (直接或者是间接都可以)。Tyvj 是一个公益性的网站，没有过多的利润，所以你要尽可能的使费用少才可以。

目前你已经知道，Tyvj 的通信渠道分为两大类，一类是必选通信渠道，无论价格多少，你都需要把所有的都选择上；还有一类是选择性的通信渠道，你可以从中挑选一些作为最终管理员联络的通信渠道。数据保证给出的通行渠道可以让所有的管理员联通。

【一本通图最小生成树】联络员 (liaison)

我们考虑如果没有必选边，那么这道题就是求原图的最小生成树

现在多了这些必选边，我们考虑从题目的要求开始分析，要求是选的边权和最小，其实在保证联通的情况下，选出的选择性边的边权和最小
这就要求我们少选，且选的边的边权尽可能的小
我们发现，这其实就是 Kruskal 选边的思路

于是我们就可以先把那些必选边连上，然后在可选边中按 Kruskal 选边的方式选边，即每次选不连通最小边权的为选边
最后的图（可能不是树）的边权和就是答案

【一本通图最小生成树】联络员 (liaison)

对于 Prim
可以在连好必选边之后把所有连通块缩成一个点，然后跑最小生成树
最后的答案就是最小生成树的边权和加上所有必选边的边权和

大家可以在有空闲时间的情况下尝试一下

【一本通图最小生成树】连接格点

有一个 M 行 N 列的点阵，相邻两点可以相连。一条纵向的连线花费一个单位，一条横向的连线花费两个单位。某些点之间已经有连线了，试问至少还需要花费多少个单位才能使所有的点全部连通。

m,n<=1000

【一本通图最小生成树】连接格点

这道题其实是和上一道题等价的
区别就是必选边的边权是 0

我们将所有点和其相邻的点按照题目要求连边，然后按照上一道题的方法跑就行

【一本通提高篇最小生成树】新的开始

发展采矿业当然首先得有矿井，小 FF 花了上次探险获得的千分之一的财富请人在岛上挖了 n 口矿井，但他似乎忘记考虑的矿井供电问题……为了保证电力的供应，小 FF 想到了两种办法：

1. 在这一口矿井上建立一个发电站，费用为 v（发电站的输出功率可以供给任意多个矿井）
2. 将这口矿井与另外的已经有电力供应的矿井之间建立电网，费用为 p

小 FF 希望身为"NewBe One" 计划首席工程师的你帮他想出一个保证所有矿井电力供应的最小花费

【一本通提高篇最小生成树】新的开始

如果没有建立发电站的选项的话，这道题就是一道最小生成树的板子

现在我们想要同时满足“至少有一个点有发电站”和“可以通过建立发电站或向已经通电的点连电线的方式覆盖电力”这两个条件

对于“发电站”这个特殊的条件，我们有两种不同的处理方式

【一本通提高篇最小生成树】新的开始

第一种，也是一种比较通用的方法

这个点有发电站，也就是说这个点不需要与其它所有点联通；换一个视角来说，也就是这一个点已经和所有点联通了
同时，我们还需要至少一个点有发电站

那么我们可以建立一个超级源点，然后把所有点和超级源点之间连一条边权为 v 的边，这样就可以同时满足“至少有一个点有发电站”和“建立发电站或连电线来覆盖所有点”这连个条件了

这种建立源点的方法也是一种比较常见的处理这种建立 + 覆盖的问题的一种方法

【一本通提高篇最小生成树】新的开始

第二种，针对于 Prim 的想法

对于这道题来说，每个点有两种选择，一种是和别的连通块联通；另一种是建立发电站，让自己成为一个新的连通块

而这正好和 Prim 求最小生成树的思路不谋而合

对于建立发电站，我们的处理方法是，将每个点的 dis 初值设置为 v，这就相当于不拉电线而直接建立发电站
而对于拉电线，我们正常建点与点之间点边
然后最后整体跑 Prim 即可

一本通提高篇最小生成树】新的开始

注意，这道题不能先跑最小生成树，后断边

如果先断边，我们无法确定发电站应该建立在子树的哪个点上

如果先选点，我们无法确定我们应该断祖先的哪条边

【一本通提高篇最小生成树】Tree

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

题目保证有解。

【一本通提高篇最小生成树】Tree

这道题需要处理的东西其实只有一个，也就是怎么控制白边的数量

我们思考 Kruskal 的算法流程，我们会发现在边的数量一定的情况下，边权更小的边会被优先加入生成树
这会启发我们思考能不能在不影响最终答案的情况下用边权来控制白边的数量

事实上，我们会发现，我们可以把所有白边加上一个权值，然后在求完最小生成树后减去这个权值乘上选取白边的数量

现在的问题就变成了，如何控制这个权值，使得我们能刚好加 need 条白边

我们会发现一个有趣的性质，也就是最终生成树的白边数量和我们加的权值是单调相关的，权值越大，最终加的边也就越少
这使得我们可以二分这个权值，然后对于二分出来的每个权值跑最小生成树，最后确定我们正确的权值，然后在生成树中减去这个权值对应的数值

【一本通提高篇最小生成树】次小生成树

给定一个无向连通图，求一颗生成树 T，使得该生成树的边权和严格大于最小生成树的边权和，且大于等于其它所有的生成树

【一本通提高篇最小生成树】次小生成树

我们考虑一张图除了最小生成树内的边之外，剩下的边有什么作用

对于多余的边 i，如果在最小生成树上加入这条边，那么就会生成一个且只有一个环
我们再删掉除了 i 以外的边，一定会构成一个生成树，且一定比原来的最小生成树边权和大

那么我们希望这棵生成树尽可能小，所以我们应当删除除了 i 以外最大的边

我们对于每一条没有选入最小生成树的边都进行这个操作，即找这个环上边权最大的边
然后对于每一条边得到的生成树求最小值

【一本通提高篇最小生成树】次小生成树

对于这个环我们发现，它的大部分都是连续的树边
或者说，只有这条多余边 i 不是树边，而是连接树上一条路径两端的一条边
我们要删除的边也是在这条路径上的最大值

于是，我们就可以用树上倍增来求这个最大值
也就是用倍增求 LCA 的方法，同时求出路径最大值

CF1051F

给定一个 n 个点 m 条边的无向连通图, 点 u_i 与点 v_i 之间有一条长度为 d_i 的边

有 q 次询问, 第 i 次询问回答从 u_i 到 v_i 的最短路长度

$1 \leqslant n, m \leqslant 10^5, m - n \leqslant 20$

$1 \leqslant u_i, v_i \leqslant n, 1 \leqslant d_i \leqslant 10^5$

CF1051F

我们注意到，$m-n$ 很小，这了给我们操作的空间

对原图建一棵最小生成树，我们发现非树边的数量最多只有 21 个
我们将非树边所连接的点称为" 特殊点"，会发现这样的点最多只有 42 个
我们将特殊点的集合成为 SN

对于两个点 u, v，他们之间的最短路 $dis(u, v)$ 要么是全走树边，要么经历一些非树边
对于只走树边，直接求 lca 然后求距离即可
对于走非树边，我们考虑枚举特殊点，那么显然有

$$dis(u, v) = \min_{k \in SN} dis(u, k) + dis(k, v)$$

我们只需要预处理出来这些特殊点到其它点的最短路即可，用 Dijkstra 实现
最后把两种情况取 min 即可

【一本通提高篇最小生成树】最小生成树计数

现在给出了一个简单无向加权图。你不满足于求出这个图的最小生成树，而希望知道这个图中有多少个不同的最小生成树。(如果两颗最小生成树中至少有一条边不同，则这两个最小生成树就是不同的)。
由于不同的最小生成树可能很多，所以你只需要输出方案数对 31011 的模就可以了。

注意：具有相同权值的边不会超过 10 条

【一本通提高篇最小生成树】最小生成树计数

这道题需要用到最小生成树的两个性质

1. 一个图的所有最小生成树中的每种边权的边的数量都是一样的
2. 无论某个边权的边的选择方式是如何的，图的联通性不变

也就是说，只要我们每种边权选出固定数量条，且满足不出环的条件，就能构成最小生成树
由乘法原理可知，我们只需要知道每种边权可行的选择方案，然后将这些方案数乘起来就行

我们按照 Kruskal 的顺序从小往大加边，由于不同边权的边的数量很少，我们可以暴力 dfs 出我们选了那些边
由于性质 2 的存在，使得对于小的边权加完之后，大的边的加边方式不会被影响

因为要统计某个边权的边的加法，dfs 过程中需要反悔操作，也就是把上一步合并的连通块断开
所以并查集不能使用路径压缩，也就是直接连接这两个点

【一本通提高篇最小生成树】最小生成树计数

这里提供另一种思路
由于最小生成树种每种边权的边的数量是一样的，我们考虑对于某一个边权，先把别的边权的边加上，然后再回头加这个边权的边
由于性质 2，当前图的连通性是固定的，连通块内部的边没有意义，我们就可以把他们缩成一个点
而我们要做的，其实就是通过这个边权的边，把这些连通块变成的点连成一棵生成树

现在，我们只需要统计每种边权所能构成的生成树数量
这里需要用到生成树计数，大家感兴趣可以自行了解一下

最短路径

最短路径顾名思义，即是求一张图上两个点之间所有路径中边权和最小的一条路径

一般来说，最短路问题有全源最短路和单源最短路

对于全源最短路，我们将介绍 Floyd
对于单源最短路，我们将介绍 Dijkstra 和 SPFA

Floyd

Floyd 是一种经典的用来求全源最短路的算法，具有特别好写的优点（

当然，除了求最短路以外，Floyd 还会有其它的应用

Floyd

这个算法的过程其实就是不断在寻找中转点的过程，也就是新的中转点 $\rightarrow$ 新的路径 $\rightarrow$ 可能的新的最短路

我们令 $e[k][i][j]$ 表示当前已经使用了编号为 $1 \sim k$ 的点为中转点，点 i 与点 j 之间的最短路

现在我们考虑如何更新下一个中转点 $k+1$ 的情况
对于新情况下的点 i 和点 j，它们有两种选择：

1. 按照原来的方式走，即 $e[k+1][i][j] = e[k][i][j]$
2. 经过新的中转点，走一条全新的路径，即
   $e[k+1][i][j] = e[k][i][k+1] + e[k][k+1][j]$

我们肯定是从这两种方式中选取最优解
于是我们就得到了求任意两点 i，j 之间的最短路，即
$e[k+1][i][j] = max(e[k][i][j], e[k][i][k+1], e[k][k+1][j])$

时间复杂度为 $O(n^3)$
空间复杂度为 $O(n^3)$

Floyd

我们观察整个算法流程会发现，我们其实没有必要保留 $1 \sim k-1$ 层的信息，因为他们其实都被整合到第 k 层内了

同时我们会发现，在更新第 $k+1$ 层时，其所需的第 k 层的信息并不会被覆盖

1. 对于 $e[k][i][j]$，本来当前就要更新它，不存在覆盖的问题
2. 对于 $e[k][i][k+1]$ 和 $e[k][k+1][j]$，如果被更新到了，也就只有两种可能：要么是通过 $1 \sim k$ 的点中转，要么时 i 或 j 与 $k+1$ 有直接连边，不可能存在通过 $k+1$ 中转的问题，也就不会被覆盖

于是我们就可以把数组的最外层压掉，即

$e[i][j] = max(e[i][j], e[i][k+1], e[k+1][j])$

注意，以上的操作没有变动原有的三层循环，也没有变动他们的顺序，单纯是空间上的优化

时间复杂度 $O(n^3)$

空间复杂度 $O(n^2)$

Floyd

对于这种找中转点的思路，我们还可以将其应用到求别的东西上

例如，求一张图上任意两点是否连通，就可以在 Floyd 上稍作改动
即 $e[i][j] = e[i][k+1] \& e[k+1][j]$
我们的数组也就只用存储连通性，也就可以用 *bool* 数组，这给了我们利用数据结构优化时间的可能

我们可以将 e 数组开成 $bitset < N > e[N]$，一种可以更快进行整行操作的 *bool* 数组
而判断 $e[i][k+1]$ 的连通性后
更新过程就可以变成 $e[i] = e[j]$，且不在需要枚举 j
时间复杂度变为 $O(\frac{n^3}{\omega})$

Floyd

除此之外，我们还可以利用这个思路解决最小环问题

我们考虑一个环最少为三个点
那么我们就可以把任意一个环拆成三部分，即 $i-j, i-k, j-k$
寻找最小环，即让这三个部分的和最小

对于 Floyd 枚举的当前层，在没有以 k 为中转点更新最短路之前，
$e[i][j], e[i][k], e[k][j]$ 三条路径恰好构成一个环
且由于这三个路径都是以 $1 \sim k-1$ 的点为中转点的情况下的最短路，
那么这个环也就是当前包含这三点的最小环

对于每一个中转点 k，显然都可以有这个操作
随着 Floyd 的进行，我们可以同时求出最短路和以 k 为断点的包含 i, j 的最小环

所有求出的环的最小值也就是全局的最小环

Dijkstra

Dijkstra 和 SPFA 的核心都在于松弛操作，即
$dis[v] = min(dis[v], dis[u] + w)$
这个操作是显然的，新更新的节点 u 也只有可能会直接影响其相连的点 v 的最短路，我们尝试使用 $S \to u \to v$ 这一条新的路径更新点 v 的最短路
感性理解下，相当于一层一层扩展，每一次松弛操作会将当前最短路的范围向外扩展一层
因为后面的节点一定会被前面的节点更新，所以我们在松弛过程中也没有必要更新后面的节点

设源点为 s，$dis[i]$ 为当前源点到点 i 的最短路
我们考虑当前已经更新了集合 E 中的点的最短路，现在已经更新了点 e 的最短路，考虑如何更新其它点的最短路
我们取出当前还未进行松弛过的点中离源点最近的点 u，然后对这个点进行松弛操作，每一个点只会松弛一遍
时间复杂度 $O(n^2)$

Dijkstra

这种做法的正确性在没有负边权的图上是可以保证的
我们考虑当前要进行松弛操作的点 u，和另一个还未进行松弛操作的点 u'，他们都和点 v 之间有边
那么在松弛点 v 之前，我们考虑源点到点 v 之间最短路的情况，困扰我们的其实只有 $S \to u \to v$，$S \to u'$，和 $S \to u' \to v$ 这三条路经的关系

1. 如果 $S \to u \to v$ 这个路径最短，且小于 $S \to u'$，由于这条到 u 的路径已经是最短路，所以这条到 v 的路径就是最短路。同时，更新过的 $S \to v$ 会优先于 $S \to u'$ 被松弛，v 点的最短路因此固定
2. 如果 $S \to u \to v$ 这条路径长于 $S \to u'$，那么就会在松弛点 v 之前松弛点 u'，保留了从点 u' 更新最优解的可能性

这样我们就说明了 Dijkstra 的正确性

在存在负边权的情况下，可能会在第一个条件出错，即 $S \to u \to v$ 要小于 $S \to u'$，点 v 的松弛会在点 u' 之前
但存在一种可能，$u' \to v$ 负边权的存在，使得 $S \to u' \to v$ 的距离小于 $S \to u \to v$，但由于点 v 已经松弛，也就无法更新到更优的解

Dijkstra

我们考虑如何优化这个算法
其实，整个算法的瓶颈在于如何维护这个当前未松弛的，到源点距离最短的点
维护这个东西，需要我们有一个容器，满足能够插入新的未松弛的点，快速查找当前最小的未松弛的点，将这个点从未松弛点的容器中删掉

我们考虑用堆来维护这个东西

每次操作从堆中取出堆顶元素，判断该点是否已经松弛
若没有进行松弛，则用它进行松弛操作，并将新的点加入堆
最后将该点从堆中弹出
时间复杂度为 $O(mlogm)$

SPFA

SPFA 其实是针对 Bellman–Ford 进行队列优化的算法，可以用来处理负边权问题

相较于 Dijkstra，Bellman-Ford 采取的松弛策略是对整张图所有点进行松弛，当在一次的操作中没有松弛成功的点时就停止
由于最短路的边数一定小于 $n-1$，所以我们最多会做 $n-1$ 轮松弛

若一张图存在负环，即边权和为负数的环，那么这个环上的点的最短路每转一圈都会变优
也就是说，整体的循环次数会多于 $n-1$
我们利用这个性质来判断负环

而由于我们是对整张图进行松弛，所以也不会存在负边权影响更新的问题，也就可以正常处理存在负边权的图

SPFA

我们会发现，其实我们没有必要松弛所有的点
会被影响的点，只有一次松弛操作相连的点

我们使用队列来存储每一次松弛过的点，这样可以优化整体的松弛次数
同时我们保留了所有点的松弛机会，也就不存在 Dijkstra 中 u' 的情况，
我们仍然可以处理负边权图

虽然在大部分情况下，SPFA 跑的很快
但是可以构造出数据（如网格图 + 菊花），可以造成反复松弛的情况，
从而使整体的复杂度降低到 $O(nm)$
所以应当慎用该算法

【一本通图最短路径算法】最小花费

在 n 个人中，某些人的银行账号之间可以互相转账。这些人之间转账的手续费各不相同。给定这些人之间转账时需要从转账金额里扣除百分之几的手续费，请问 A 最少需要多少钱使得转账后 B 收到 100 元。

【一本通图最短路径算法】最小花费

这道题其实就是相当于找一条转账的路径，使得最后剩下的百分比尽可能的多
需要注意的是，不能求扣除的比例最小值，因为扣除的是前一次剩下钱的比例，并不是扣除钱的一定比例，所以不能直接相乘

求解过程其实就是将最短路改成最长路，且松弛的过程从加法变成乘法

【一本通图最短路径算法】香甜的黄油

农夫 John 发现做出全威斯康辛州最甜的黄油的方法：糖。把糖放在一片牧场上，他知道 N（1<=N<=500）只奶牛会过来舔它，这样就能做出能卖好价钱的超甜黄油。当然，他将付出额外的费用在奶牛上。
农夫 John 很狡猾。像以前的 Pavlov，他知道他可以训练这些奶牛，让它们在听到铃声时去一个特定的牧场。他打算将糖放在那里然后下午发出铃声，以至他可以在晚上挤奶。
农夫 John 知道每只奶牛都在各自喜欢的牧场（一个牧场不一定只有一头牛）。给出各头牛在的牧场和牧场间的路线，找出使所有牛到达的路程和最短的牧场（他将把糖放在那）

牧场数（2<=P<=800），牧场间道路数 C(1<=C<=1450)

【一本通图最短路径算法】香甜的黄油

由于这道题点数和边数很少
我们考虑枚举牧场求单源最短路，然后计算奶牛到这个牧场的距离和，最后求最小值

【一本通图最短路径算法】信使

战争时期，前线有 n 个哨所，每个哨所可能会与其他若干个哨所之间有通信联系。信使负责在哨所之间传递信息，当然，这是要花费一定时间的（以天为单位）。指挥部设在第一个哨所。当指挥部下达一个命令后，指挥部就派出若干个信使向与指挥部相连的哨所送信。当一个哨所接到信后，这个哨所内的信使们也以同样的方式向其他哨所送信。直至所有 n 个哨所全部接到命令后，送信才算成功。因为准备充足，每个哨所内都安排了足够的信使（如果一个哨所与其他 k 个哨所有通信联系的话，这个哨所内至少会配备 k 个信使）。
现在总指挥请你编一个程序，计算出完成整个送信过程最短需要多少时间。

【一本通图最短路径算法】信使

我们考虑其实每个点在第一次收到信息之后就会往下传播信息，后续在收到的信息也就没有意义
所以其实每一个点收到信息所需的时间就是源点到这个点的最短路
那么所有点都收到信息的时间就是源点到所有点的最短路的最大值
注意判断无解情况，也就是没有最短路

【一本通图最短路径算法】最优乘车

H城是一个旅游胜地，为方便游客，在各个旅游景点及宾馆、饭店等地设置了 N 个巴士站，并开通了 M 条一些单向巴士线路。
现在用整数1， 2， …， n给H城的所有巴士站编号，约定CUP饭店的巴士站编号为1， S公园巴士站的编号为N。
写一个程序，寻找一个从CUP饭店到S公园的过程中换车的次数最少的乘车方案。

1<=M<=100 1<N<=500

【一本通图最短路径算法】最优乘车

这个问题需要考虑的点就是，如何能够使换乘的信息和沿着一条线路一直走的信息共同存在于同一个图中
如果我们只用一张图，显然无法处理在同一点换成的问题

我们考虑将每一个点拆开，拆成 $1 \sim M$ 个点，分别表示每一条巴士线路
我们将在点 i 处有停靠站的巴士线路表示的点之间连边，边权为 1，走这些边就代表换乘了
同时，在每一条巴士线路内部按照线路连边权为 0 的边，这些边表示不换乘，沿着一个线路走

最后只要能到达点 n，无论是哪个线路的都是合法的
为了方便，我们将每一个线路的点 n 都连向一个超级汇点 T，将所有点 1 连向一个超级源点 S
问题就转化成了 S 到 T 的最短路

这就是很典型的拆点，或者说分层图的思路

【一本通提高篇最短路】最优贸易

现在有 n 个点，m 条边，这 m 条边有一部分是单向的，有一部分是双向的
现在可以在这 n 个点买卖水晶球，第 i 个点水晶球的价格是 a_i

现在小 L 从点 1 走到点 n，它可以选取一个点购入水晶球，并选取一个点卖出水晶球
求小 L 可以赚取的最高差价

$1 \le n \le 100000, 1 \le m \le 500000, 1 \le a_i \le 100$

【一本通提高篇最短路】最优贸易

我们考虑如何处理这个买入和卖出
很显然买完肯定要卖，不卖肯定亏

根据上一道题的提示，我们考虑将整张图分成三层，即没买水晶球，买完没卖水晶球，卖了水晶球
一层和二层之间的边权是负的价格，表示买入
二层和三层之间是正的价格，表示卖出

最后的答案就是第一层的点 1 到最后一层的点 n 之间的最长路

【一本通提高篇最短路】最优贸易

第二种思路

我们其实没有必要关心在哪个点买，然后在哪个点卖
只要有一个点买入有一个点卖出就行了
范围再大一点，只需要在某个区间内的点买入，再在某个区间内的点卖出就行
这就又回到了 Floyd 的时候思考过的中转点的思路

我们考虑枚举点 i 为中转点，如果想要最大收益，肯定是在 $1->i$ 的路径中选一个最小点买入，在 $i->n$ 的路径中选一个最大点卖出即可
前者是最短路，后者是最长路
最终答案就是所有 i 的收益的最大值

由于有负环，要用 SPFA
这道题是点权，而且只选择一个点买入或者卖出，所以要改变一下 SPFA 的实现方法，变成 $dis[v] = min(dis[v], dis[u])$
$dis[u]$ 表示从源点到 u 的最小点权

【一本通提高篇 SPFA 算法的优化】Wormholes 虫洞

John 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）。John 的每个农场有 M 条小路（无向边) 连接着 N （从 1..N 标号）块地，并有 W 个虫洞。其中 1<=N<=500,1<=M<=2500,1<=W<=200。现在 John 想从某一个农场出发，借助这些虫洞来回到这个农场的过去(从这个农场出发时刻之前)，请你告诉他能办到吗。John 将向你提供 F(1<=F<=5) 个农场的地图。没有小路会耗费你超过 10000 秒的时间，当然也没有虫洞回帮你回到超过 10000 秒以前。

【一本通提高篇 SPFA 算法的优化】Wormholes 虫洞

这道题其实就是裸的判断负环问题

对于 SPFA 来说，每一条边最多会参与松弛操作 $n-1$ 次
也就是说，如果某条边访问超过 $n-1$ 次，那么就出现了负环

CF2023B

小 Z 有 n 道题目，第 i 道题目有 a_i 和 b_i 两个属性
当小 Z 决定挑战第 i 道题目时，他有两个选择

1. 做这道题，则小 Z 会获得 a_i 分，且小 Z 选择的下一道题目 j 需要满足 $j < i$ 且没有挑战过
2. 跳过这道题，则小 Z 获得 0 分，且小 Z 选择的下一道题目 j 需要满足 $j \leqslant b_i$ 且没有挑战过

小 Z 将从 1 号题目开始挑战，并想追求更高的分数
现在小 Z 想问问聪明的你，他最高能够获得多少分？

$1 \leq n \leq 4 \times 10^5$

我们先考虑第一个条件
如果小 Z 能够到达第 k 道题，那么所有编号小于等于 k 的没挑战过的题都能得分
同时我们发现，想要做后面的题目，必须要跳过前面的题目
所以我们只需要对于每一个题目，求出想要到达这道题需要跳过多少题就行了

现在我们想要得分尽可能的多，在没跳过的题目的分都可以得到的情况下，我们肯定是想要跳过题的分数越小越好
题目就被转化为了，到达第 k 个题目需要浪费的最小分数

CF2023B

我们发现，题目与题目之前其实是“到达”或者说“继承”的关系，到达同一个题目还会有很多种方法，这启发我们能不能建图去做

对于每一个 i，我们向 $i-1$ 连一条边权为 0 的边，代表第一个条件
同时对于每一个 i，我们向 b_i 连一条边权为 a_i 的边，代表跳过这道题，会浪费 a_i 的分数
那么到达第 k 道题浪费的最小分数就是 1 到 k 的最短路
以 1 为源点求最短路，每道题的答案就是，这个点之前所有点的分数和减去到达这个点浪费的分数

最后对于每个点的答案求最大值

P4568

给定一个 n 个点 m 条边的无向连通图, 点 u_i, v_i 之间边的边权为 w_i
可以使 k 条边的边权变成 0, 求点 s 到点 t 之间的最短路
$2 \leqslant n \leqslant 10^4, 1 \leqslant m \leqslant 5 \times 10^4, 0 \leqslant k \leqslant 10$
$0 \leqslant s, t, u_i, v_i < n, 0 \leqslant w_i \leqslant 10^3$

P4568

我们显然把这 k 次都用完是最优的
我们考虑将图分层
设 i 为当前层数
层内对于一条边 $u \to v$, 正常连一条 (u_i, v_i, w) 的边
对于层之间, 我们连接 $(u_i, v_{i+1}, 0)$, 表示删掉了这条边的边权, 且用掉了一次次数
最终答案就是 $dis[t_k]$
注意节点边号与实际数组下标换算的问题

【一本通提高篇最短路】Telephone Lines 架设电话线

Farmer John 打算将电话线引到自己的农场，但电信公司并不打算为他提供免费服务。于是，FJ 必须为此向电信公司支付一定的费用。FJ 的农场周围分布着 $N(1 <= N <= 1,000)$ 根按 $1 \sim N$ 顺次编号的废弃的电话线杆，任意两根电话线杆间都没有电话线相连。一共 $P(1 <= P <= 10,000)$ 对电话线杆间可以拉电话线，其余的那些由于隔得太远而无法被连接。第 i 对电话线杆的两个端点分别为 A_i、B_i，它们间的距离为 L_i $(1 <= L_i <= 1,000,000)$。数据中保证每对A_i，B_i 最多只出现 1 次。编号为 1 的电话线杆已经接入了全国的电话网络，整个农场的电话线全都连到了编号为 N 的电话线杆上。也就是说，FJ 的任务仅仅是找一条将 1 号和 N 号电话线杆连起来的路径，其余的电话线杆并不一定要连入电话网络。经过谈判，电信公司最终同意免费为 FJ 连结 $K(0 <= K < N)$ 对由 FJ 指定的电话线杆。对于此外的那些电话线，FJ 需要为它们付的费用，等于其中最长的电话线的长度（每根电话线仅连结一对电话线杆）。如果需要连结的电话线杆不超过 K 对，那么 FJ 的总支出为 0。请你计算一下，FJ 最少需要在电话线上花多少钱。

【一本通提高篇最短路】Telephone Lines 架设电话线

首先看到 K 条免费线路，我们想能不能用分层图做

第 i 层图代表使用了 $i-1$ 次免费机会
对于原图的每一条边，本层内按照正常的边权连边，并且向下一层连接一条边权为 0 的边

最后用求第一层 1 号点到最后一层 n 号点的最短路
注意最短路的实现方式

【一本通提高篇最短路】Telephone Lines 架设电话线

另一种思路，我们可以参考那倒规定数量白边最小生成树的题目
我们思考，能不能通过边权来控制使用免费机会的次数

我们考虑固定一个边权最大值 $maxs$
那么问题就变成了，存不存在一个方案，最多走 K 条边权大于 $maxs$ 的边，从点 1 到点 n
即寻找一条“走边权大于 $maxs$ 的边数量最少的路径”

我们考虑将所有边权大于 $maxs$ 的边的边权变成 1，其余的边权变成 0
那么我们走的最少的“边权大于 $maxs$ 的边”的数量就是在这张新图上的 1 到 n 的最短路

由于我们只有 K 次免费机会，也就是说，最多走 K 条“边权大于 $maxs$ 的边”
我们用这个判断当前这个最大边权是不是合法的

同样的，这个最大边权和使用的免费次数还是单调相关的，我们就可以二分这个最大边权

广义来说

很多图论题其实并没有“路径”之类的提示

更多情况需要从问题中分析出不同元素之间的关系

把元素抽象成点，把关系抽象成边，最后把一个问题抽象成在图上操作的图论问题

谢谢大家

QQ:10066692