图

wx

T1

问题描述：

N 个点 M 条边的有向图，边有黑白两种颜色。现在要给点染色，每个点染成黑或白色。

白点只能走它连出去的白边，黑点只能走它连出去的黑边。

问是否存在一种染色方案，使得**不存在**一条 $1 \to n$ 的路径。如果存在这样的染色方案，在第一行输出 -1，否则输出 $1 \to n$ 最长的最短路径长度，每条边长度为 1 。在第二行，输出对应第一行答案的染色方案。

$1 \le N , M \le 500000$

T1

问题分析:

- 从 1 出发染色不好处理，考虑从 n 出发进行染色，尽可能让每一条路不可经过，这样也是最大化其他点到 n 的最短路。
- 如果当前为 u，点 v 和 u 有边，如果只有一种颜色的边，那么这条路是可以禁止经过的，将 v 设置成与边不同的颜色。如果有不同颜色的边，那么 v 的颜色无论怎么染色都可以到达 u 。

T1

问题分析：

- 从 n 开始进行反向 BFS。
- 当第一遍历到未染色的点 x，将点 x 设为与边不同的颜色。如果遍历到染色的点，说明此点无法避开，加入到队列中，更新到 n 的最短路，直到 1 入队。
- 时间复杂度为 O(N+M)。
- 来源：CF1407E Egor in the Republic of Dagestan

T2

问题描述：

在一个有 N 个顶点和 M 条边的图上有两个人，分别在 S 号节点和 T 号节点。他们要各自走到对面（即在 S 的人走到 T，在 T 的人走到 S）。

给你 M 条边，描述为 U V D 分别表示该边连接的两个点及通过边的时间。

求两人经过最短路径（可能有多条）且不相遇（在同一单位时间内都在一条边或一个点上）的方案数。

$N \le 100000$，$M \le 200000$

T2

问题分析：

- 直接求解比较麻烦，考虑用所有方案数减去非法的方案数得到答案。
- 首先求解出 S,T 出发到任意点 x 的最短时间 $F_S[x],F_T[x]$ 和路径数量 $G_S[x],G_T[x]$。
- 不考虑相遇的情况，方案数为 $G_S[T]\times G_T[S]$。

T2

问题分析：

- 考虑相遇的情况，两人相遇肯定只会相遇一次。
- 第一种，在某个点 u 上相遇，枚举相遇的点，需要满足：
  - $F_S[u]=F_T[u]$
  - $F_S[u]+F_T[u]=F_S[T]=F_T[S]$
- 方案数为 $(G_S[u]\times G_T[u])^2$。

T2

问题分析：

- 第二种，在边 (u,v) 上相遇，枚举相遇的边，需要满足：
  - $F_S[u]+F_T[v]+d=F_S[T]=F_T[S]$
  - $F_S[u]+d>F_T[v]$
  - $F_T[v]+d>F_S[u]$
- 方案数为 $(G_S[u]\times G_T[v])^2$。
- 时间复杂度为 O(Nlog N)。
- 来源：[ARC090C]Avoiding Collision

T3

问题描述：

有一个地铁线路图，可以看做 N 个站点， M 条线路。每条线路由一个公司所有。

如果你乘坐同一公司的铁路，只需要花费 1 元，如果更换其他公司铁路，还需要再花费 1 元，如果再次换回原来的公司，还需要花费 1 元。

问从 1 号站点到 N 号站点的最小花费。

$N,M \le 200000$

T3

西南大学附属中学
High School Affiliated to Southwest University

问题分析：

- 将同一公司相连的路线看做一个联通块，联通块内任意节点直接的花费均为 1 。
- 最朴素的想法就是将联通块中任意两点之间连一条边权为 1 的边，然后直接跑最短路。
- 但这样建图边数太多，最坏情况所有点在同一个联通块内，有 $n(n-1)/2$ 条边，超空间了。

T3

问题分析：

- 对于每个联通块，建立一个虚点，联通块上的边全部删除，每个点向虚点连边，边权为 0.5，这样，联通块上的任意两点都可以通过虚点作为中间点通过边权和为 1 的方式连接起来。
- 最坏情况下每条边上的两个点为一个联通块，最多 n+m 个点，2m 条边。
- 建图后直接跑最短路。
- 来源：[ARC061C]Snuke's Subway Trip

T4

问题描述:

现在有 N 个点，编号 1~N，起点为 s。

现在你有 Q 种连边方式，方式分为三类:

- 连接 u,v， 花费为 w。
- 将 u 和编号 [l,r] 之间的所有点连边，花费 w。
- 将编号 [l,r] 之间的所有边连边，花费 w。

问：起点到所有点的最小花费。

$1 \le N,Q \le 10^5$

T4

问题分析：

- 直接建边会超空间。
- 对于区间，可以想到线段树，使用**线段树优化建图**。

T4

问题分析：

- 先考虑第二种连边方式。
- 首先建立一棵线段树，每个节点表示一个区间，区间到孩子节点连一条有向边，边权为 0，因为已经包含了。
- 如果点 u 要与区间 [l,r] 相连，则将节点 u 连接到 [l,r] 对应区间的节点上，可能有多个节点，边权为 w。

T4

西南大学附属中学 High School Affiliated to Southwest University

问题分析：

- 对于第三种连边方式，区间向节点连边，用类似的方式。
- 建立一棵线段树，孩子节点到父亲节点连一条有向边，边权为 0。
- 如果区间 [l,r] 要和节点 u 相连，将 [l,r] 对应区间的节点上连到 u 上，边权为 w。

T4

问题分析：

- 如果同时存在第二种第三种，建立两棵线段树。
- 一棵线段树存在父亲到孩子节点边权为 0 的有向边，一棵线段树存在孩子节点到父亲边权为 0 的有向边。
- 叶子节点就是单点，可以直接连边。同时，两棵线段树的同一位置的叶子节点连一条边权为 0 的边，因为它们是同一个点。
- 建图后跑最短路。
- 来源：CF786B Legacy

T5

问题描述：

给定 n 个点 m 条边的无向图。

有 q 次询问，每次询问 k 条边，是否能同时在最小生成树上。

$n,m,q,\sum k \le 500000$

T5

问题分析：

- 所有最小生成树中，每种权值的边的数量是一定的。
- 先考虑一组询问的情况。
- 将询问的边权从小到大排序，对于当前边权 w， 求解最小生成树树，确保已经优先考虑了 < w 的边，再尝试加入当前边权 w， 如果有多个相同的 w 则同时加入，如果有一条加入失败，则不成立。
- 因此，同一组询问，不同边权的边不会互相影响。可以将每一个询问分成若干组边权相同的询问。

T5

问题分析：

- 对于 Q 组询问，进行离线操作，整体看作一个询问，按照一个询问的方式求解。
- 当前一组询问结束后，需要撤销，这里需要用可撤销并查集求解最小生成树。
- 时间复杂度为 O(mlog m)。
- 来源：CF891C Envy

T6

问题描述：

现在有 n 个人，并且给出他们的年龄 a。

两个人是朋友，当且仅当两个人年龄的按位与结果为 0 。

现在，有一个传销组织，每个人有两种操作：

1、主动加入传销组织，这样的话，传销组织不会给你钱；

2、邀请自己的一个朋友加入传销组织，这样的话，传销组织会奖励你数值等于你的年龄的钱。（当然，执行该操作的人必须已经进入传销组织了）

每个人只可以进入传销组织一次。

现在，请你输出，如果 n 个人通力合作，传销组织支付给这 n 个人的钱数之和最大是多少。

$n,a \le 200000$

T6

问题分析：

- a_u,a_v 是朋友，建立 $a_u \rightarrow a_v$，$a_v \rightarrow a_u$ 的边，边权为 a_u，a_v，最终会形成若干个图。
- 题目求解的就是最大外向森林。

T6

问题分析：

- 建立一个节点 0， 连接所有节点。问题就转换为了求解 0 为根的最大外向树。
- 每个点作为出边都会被选一次，将 a_u,a_v 的边权设为 a_u+a_v， 题目转化为最大生成树-$\sum a_i$。
- 直接建图连边，边数 $O(n^2)$ 级别，无法直接求解。

T6

问题分析：

- 模拟生成树求解过程。
- 逆序枚举可能的边权 S，求解是否存在能组成边权 S 的 $a_u,a_v(S=a_u+a_v)$，如果存在则加入到生成树中，用并查集维护。
- 是否找出是否存在这样的 a_u,a_v 呢?
- 枚举 S 的子集 x，看 x 和 x xor S 是否存在。
- 将点权相同的可以缩点处理，例如 a_u 的数量为 cnt_{au}，a_v 的数量为 cnt_{av}。
- 则产生的贡献为 $S\times(cnt_{au}+cnt_{av}-1)$。
- 来源：[CF1305G]Kuroni and Antihype

T7

问题描述：

一个国家有 n 个城市和 m 条单向道路，一个旅行商在这些城市之间旅行。

第 i 条道路从城市 a_i 到城市 b_i，只有当他的资产不少于 r_i 元才可以走这条道路，走过这条道路之后他的资产会增加 p_i 元。

他希望自己可以永远不停的游走下去，于是他想知道从任意一个城市出发至少需要多少元初始资产。

$n,m \le 200000, 0 \le r_i,p_i \le 10^9$

T7

问题分析：

- 对于没有出边的点，肯定无法一直走下去。
- 直接考虑每一个城市 u， 经过的每一个城市都有可能更新 u， 时间复杂度太高。

T7

问题分析：

- 对于 r 最大的一条边 a,b，如果在 a 的初始资产为 r，那么肯定可以一直走下去，因为 p>=0。
- 当一个点 v 的答案确定后，它的出边都可以删除，此时出度为 0。同时可以更新起点 u 的答案 $ans_u=\min(ans_u,\max(r_{u\to v},\ ans_v-p_{u\to v}))$，更新后，边 u→v 没有作用了，可以直接删除，如果 u 的出度也变为 0，那么就意味着 u 的答案也确定了，按照类似拓扑序的方式求解出所有点答案。

T7

问题分析：

- 具体步骤:
- 1. 将出度为 0 的点入队。
- 2. 将边从大到小排序。取出队中的点 v，如果 v 是有效的，更新 u，$ans_u=\min(ans_u,\max(r_{u\to v}, ans_v-p_{u\to v}))$，同时，删除边 $u\to v$，u 的出度 -1，如果 u 的出度为 0，则入队。
- 3. 对于当前 r 最大的边 $u\to v$，如果此边未删除，更新 $ans_u=\min(ans_u,r_{u\to v})$，然后删除此边，更新出度，如果出度为 0，则入队。
- 4. 重复直到所有点入队。
- 时间复杂度为 O(mlog m)
- 来源：[CCO2021]Travelling Merchant

T8

问题描述:

给定 n 个点，m 条边。

定义一个点集是“宽松的”，当且仅当满足:

- 每条边都有至少一个顶点在这个点集中
- 最多只能有一条边，满足两个顶点都在这个点集中

请你找出一个“宽松的”点集。

$2 \le n \le 1000000$， $n-1 \le m \le \min(10^6, n(n-1)/2)$

T8

问题分析：

- 如果每条边只有一个点在点集中，那么原图就是一个二分图。
- 考虑一条边上两个点在点集中，那么一定存在奇环，将基环上的一条边上的两个点合并成一个点，原图就是二分图了，满足了题意。
- 这样的边需要在所有的奇环上，且不能在偶环上。

T8

问题分析：

- 接下来就需要统计每条边所在奇环和偶环的数量。
- 其实可以在二分图染色的时候就可以找环。环的形成是有树边和返祖边组成，遍历到返祖边就存在了环。统计可以使用树上差分。
- 找到这样的边后，删除，进行二分图染色，最终得到满足的点集，否则不存在这样的点集。
- 时间复杂度为 O(n+m)。
- 来源：[CF1680F]Lenient Vertex Cover