光网络业务连续性优化难题

> 时间限制: 90s 空间限制: 512MB

题目描述

光网络可以看成是一个由 N 个节点，M 条边组成的无向连通图，两个节点之间可能存在多条边，图上的每条边代表现实世界的一条光纤，每条光纤上存在 $K = 40$ 条光通道；

在光网络上初始时运行着多条光业务，一条起点为 S，终点为 T，宽度为 W 的光业务可以看作是一条从起点 S 到终点 T 的**简单路径**(即路径上不存在重复节点或重复边)，其中对于路径上的每条边，使用了其中的 W 个**编号连续**的通道；多条业务可以使用同一条边的不同通道，但是不能使用同一条边的相同通道；另外，每条业务会有对应的业务价值，用 V_i 表示第 i 条业务的业务价值；

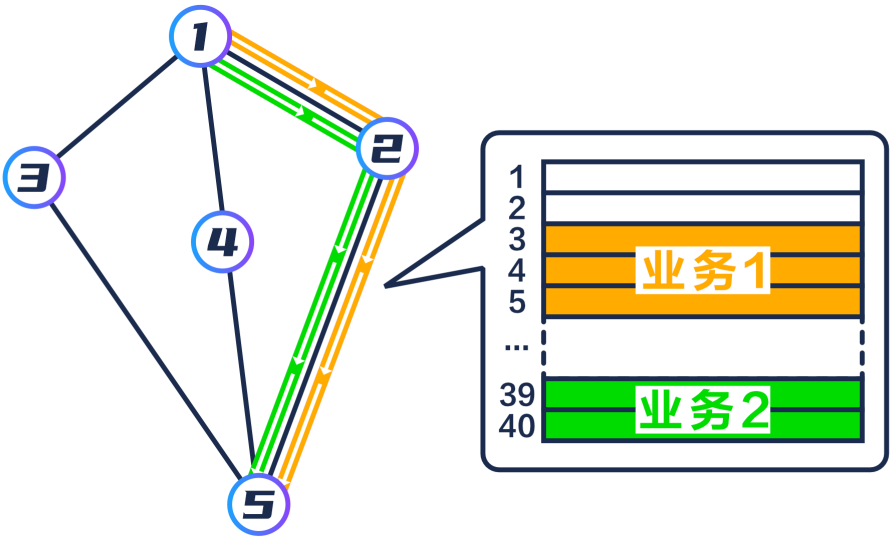

图 1

如图1所示，此时光网络中运行着两条光业务；业务 1 和业务 2 的起点为 1，终点为 5，其中业务 1 的宽度为 3，使用了通道 [3, 5]，业务 2 的宽度为 2，使用了通道 [39, 40]，它们的业务路径为 (1 -> 2 -> 5)；

在普通情形下，一条业务在其顺序经过的所有边 $e_1, e_2 \dots e_n$中，使用的通道编号集合应该是相同的；具体的，令 C_{e_k} 为该业务在路径上第 k 条边的使用通道集合，则 $C_{e_{k+1}} = C_{e_k}$； 但有些节点上存在若干次变通道的次数 P_k，令 $e_k = (u, v)$，表示这条边连接了 u 和 v 节点, $e_{k+1} = (v, w)$，如果此时节点 v 的变通道次数还有剩余 $(P_v > 0)$，则可以使用一次 P_v 使得 $C_{e_{k+1}} \neq C_{e_k}$；

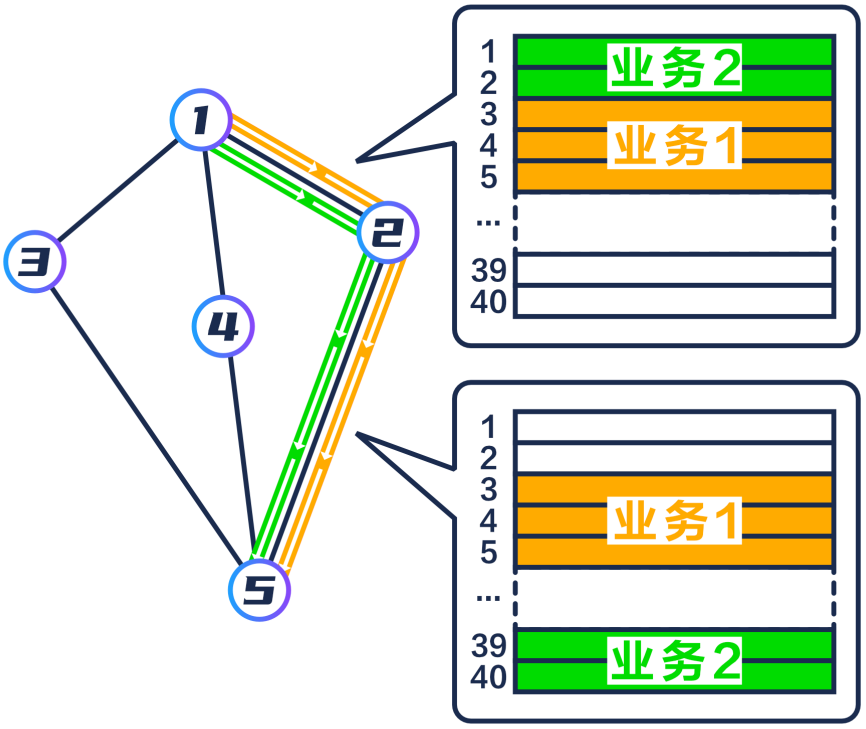

图 2

如图 2 所示，若初始时节点 2 的变通道次数为 1，可以注意到业务 2 在 (1, 2) 边上使用了通道 [1, 2]，在 (2, 5) 边上使用了通道 [39, 40]，节点 2 的变通道次数则变为 0，那么此时业务 1 则无法经过节点 2 变通道；

在实时的网络运行中，可能存在自然灾害、硬件故障或者维护活动引起的多种光纤故障情况；当光纤发生故障后，**只有**业务路径经过该故障光纤的业务需要被重新规划，但并非所有的业务都能被成功规划路径：

重新规划的业务路径必须满足，起点、终点和宽度与业务的原始属性一样，并且满足上述约束；另外，重新规划的某条业务虽然不能使用其它业务的资源，但是可以使用该业务当前的资源(即通道和节点变通道能力)，例如，如果光纤故障影响了业务 1 和业务 2，记业务 1 的新老路径为 new 1, old 1，记业务 2 的新老路径为 new 2, old 2，则 new1 不能使用 new 2 和 old 2 的资源，但 new1 可以使用 old 1 的资源。

若规划成功则业务将释放它们原来占用的边上的通道以及对应节点的变通道次数；如果 new 1 使用了 old 1 的部分资源，则该部分资源不会在规划成功后被释放。

若你没有为某些业务规划路径，则视为这些业务死亡，**它们所占用的边上的通道以及对应节点的变通道次数不会被释放，且在该测试场景下后续这些死亡的业务不能再重新规划**；

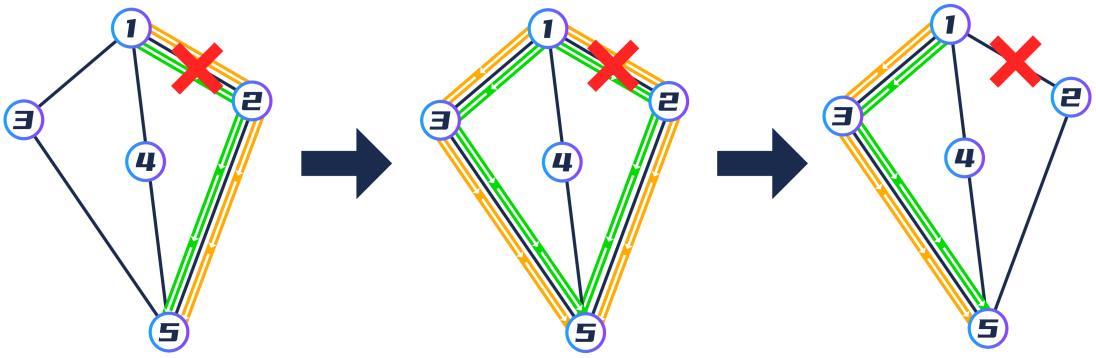

图 3

如图 3 所示，当网络发生 (1, 2) 边的中断时，业务 1 和 2 将重新规划路径，且在规划成功之后它们原始路径上占用的资源将会被恢复（路径通道，变通道次数）；

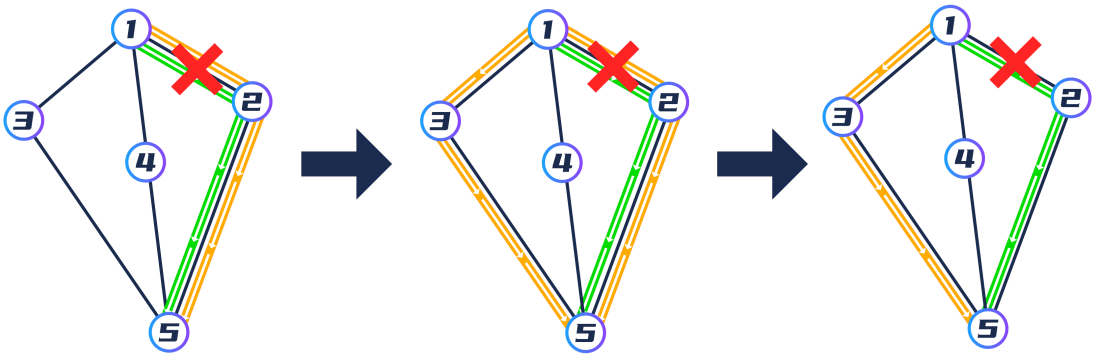

图 4

如图 4 所示，当网络发生 (1,2) 边的中断时，只有业务 1 重新规划了路径，虽然业务 2 死亡，但后续它依然将一直占用原始路径上的资源；

在这种情况下，你需要设计出一种算法，确保在多次光纤故障后，网络上存在总价值尽可能大的未死亡业务；

题目交互

你的程序应该以系统的标准输入流和标准输出流作为系统的标准输入输出；

初始环境输入

第一行两个整数 N 和 M，N 表示图的节点数，M 表示图的边数；$(2 \leq N \leq 200, 1 \leq M \leq 1000)$；

第二行 N 个整数，第 i 个整数 P_i 表示节点 i 允许的最大变通道数；$(0 \leq P_i \leq 20)$

接着 M 行每行两个整数 u_i, v_i 表示图上的第 i 条边 (u_i, v_i)；$(1 \leq u_i, v_i \leq N, u_i \neq v_i)$；

输入的图保证连通，无自环，可能有重边；

接着一行一个整数 J，表示图上初始运行的业务数；$(1 \leq J \leq 5000)$；

接着 $2J$ 行每两行表示一条业务；

每条业务的第一行六个整数 Src, Snk, S, L, R, V 表示该业务的起点为 Src，终点为 Snk，经过的边数目为 S，在路径上所有边的占用通道范围为 $[L, R]$，该业务的业务价值为 V；第二行 S 个整数依次表示业务依次经过的路径上的每条边的编号，$e_1, e_2 \ldots e_S$，其中 e_k 表示路径的第 k 条边；$(1 \leq Src, Snk \leq N, Src \neq Snk, 1 \leq L \leq R \leq K, 0 \leq V \leq 100000, 1 \leq e_k \leq M)$

注意初始业务并不会变通道；

题目中所有编号都是从 1 开始，业务的编号和边的编号按照输入顺序递增；

注意交互部分的每个测试场景，都是从该初始环境开始；

交互部分

接下来是交互部分：

首先一行一个整数 T 表示 $T, (1 \leq T \leq 100)$ 个独立的测试场景，在处理每一个场景时，都是保证**从上面的初始环境开始**的(此时还没有任何中断边)；

对于每一个场景，你的程序需要接收一系列的整数 e_{failed} ，e_{failed} 表示此时网络中发生中断的边的编号，$(1 \leq e_{failed} \leq M)$，在每接收到一个 e_{failed} 时，你的程序需要重新规划因为此次中断所影响的业务的路径，以及为路径上的每条边规划通道；具体的，对于当前的 e_{failed}，假设当前受影响的业务集合为 $Serv = \{S_1, S_2 \dots S_n\}$，那么你的程序应该输出一行一个整数 R，其中 $0 \leq R \leq n$，表示规划成功的业务数，接下来输出 $2R$ 行表示每个规划成功的业务，其中第一行两个整数 $Serv_{id}$ 和 S ，分别表示业务的编号和业务的新路径的边数，第二行 $3S$ 个整数 e_k, e_l, e_r 表示业务依次经过的边的编号以及在该边上占用的通道区间为 $[e_l, e_r]$；直到你的程序接收到一个整数 -1 时，则表示该测试场景结束，然后你需要将程序恢复到初始环境状态，进入下个测试场景；

T 个测试场景的总 e_{failed} 个数不超过 6000；

注意：在你每输出一行时，为了确保交互程序能正确读取你的输出，你需要刷新输出，其中：

C 和 C ++ 可以使用 fflush(stdout);

Java 可以使用 System.out.flush();

评分

对于每一个用例 i，令 T_i 为此用例测试场景的个数，令 $BeginV_i^j$ 为第 j 个测试场景开始时的业务总价值，令 $EndV_i^j$ 为第 j 个测试场景结束后存活的业务总价值；则该测试场景的得分为 $Score_i^j = \frac{EndV_i^j \cdot 10000}{BeginV_i^j}$；该用例的分数为所有测试场景的得分之和 $Score_i = \sum_{j=1}^{T_i} Score_i^j$；

题目的得分为所有用例的得分之和 $Score = \sum Score_i$；

分数越高排名越前；分数相同时，先提交的排名靠前；

样例

初始环境输入

```
5 6
1 1 1 1 1
1 2
2 5
```

1 4

4 5

1 3

3 5

2

1 5 2 1 20 1

1 2

1 5 2 21 40 1

1 2

交互部分

环境输入: 1

环境输入: 1

程序输出: 2

程序输出: 1 2

程序输出: 5 1 20 6 1 20

程序输出: 2 2

程序输出: 5 21 40 6 21 40

环境输入: 6

程序输出: 2

程序输出: 1 2

程序输出: 3 1 20 4 1 20

程序输出: 2 2

程序输出: 3 21 40 4 21 40

环境输入: -1

错误类型

基础错误类型

1. 代码编译错误
2. 程序异常退出 (可能原因: 运行错误，使用异常权限，输出参数比实际多，输出参数格式不对，etc...)
3. 超出时间限制 (可能原因: 交互时未使用清空流缓存命令，程序运行超时，输出参数比实际少，etc...)
4. 超出内存限制

逻辑错误类型

1. "Unknown Error" (出现时请联系大赛方)
2. "Incorrect Number of Services"（输出的业务数量不对）
3. "Incorrect Service ID"（输出的业务ID不对）
4. "Duplicate Service ID" (单次输出的业务ID重复)
5. "Unaffected Service ID" (输出未受到此次光纤中断影响的业务ID)
6. "Incorrect Number of Edges"（路径的边数不对）
7. "Incorrect Edge ID"（路径的边ID不对）
8. "Duplicate Edge ID" (路径上的边ID重复)
9. "Pass Break Edge" (路径经过了已经中断的光纤)
10. "Inconsistent Service width" (路径上的业务宽度不一致)
11. "Incorrect Channel ID"（业务所使用的通道ID不对）
12. "Cyclic Path" （路径成环）
13. "Channel Occupied Kind 1"（业务所使用的通道被占用，可能与其他业务的老路径有关）
14. "Channel Occupied Kind 2" （业务所使用的通道被占用，可能与其他业务的新路径有关）
15. "Disconnected Path" (路径不连通)
16. "Insufficient Channel Quantity"（节点变通道次数不够）
17. "Mismatched start and end" (业务新路径的起点或终点与老路径不匹配)