

# 2020牛客暑期多校训练营（第五场）

[比赛链接](#)

## A - Portal

Upsolved by nikkukun.

### 题目描述

推荐购买 [Portal](#) 以及 [《Portal 2》](#) 以加深题目理解

给一个  $n \leq 300$  点的带权无向连通图，有  $k \leq 300$  个任务，第  $i$  个任务需要从  $a_i$  移动到  $b_i$ 。当你经过某个节点时，你可以在此打开一个传送门，同一时刻最多只有两个传送门，且你可以随时随地关闭它们。

求完成所有任务的最短路径和。

### 解题思路

可以发现实际上是依次经过  $2k$  个关键点  $c_1, c_2, \dots, c_{2k}$  的任务。

官方题解提供了其他两种复杂度不对，但是具有启发意义的解法：

1.  $f(i, u, a, b)$  完成前  $i$  个任务，目前在  $u$  两个传送门在  $a$  和  $b$ 。由于转移有环，需要用 Dijkstra。
2.  $f(i, u, a)$  完成前  $i$  个任务，目前在  $u$  当前位置有一个传送门，另一个在  $a$ 。状态的化简是基于传送门一定是用的时候当场打开的观察的。

正解还可以继续精简状态。  $f(i, a)$  表示完成前  $i$  个任务且站在结束节点上，当前令一个传送门在  $a$ 。状态的化简是基于一次任务只会改变最多一次传送门的观察的。这样枚举本次任务后传送门变为  $b$  有三种转移：

1.  $c_i \rightarrow c_{i+1}$  不用传送门；
2.  $c_i \rightarrow a \rightarrow b \rightarrow c_{i+1}$  使用传送门到  $a$  然后经过  $b$  时建立传送门；
3.  $c_i \rightarrow b \rightarrow a \rightarrow c_{i+1}$  经过  $b$  时建立传送门，然后使用传送门到  $a$ 。

总时间复杂度  $O(kn^2)$

## B - Graph

Solved by nikkukun.

## 题目描述

给一个非负权的树，你可以任意加边或删边，但任意时刻要保证：

1. 连通
2. 任意环异或和为  $0$

求操作过后整棵树的最小权。

## 解题思路

不难发现将边权往小修改的操作其实很像最小生成树的过程（接一个环，去掉环上的最大边），又发现这个图实际上一直是一棵生成树，其边权  $(u, v)$  是原图中  $u$  到  $v$  的路径异或和，因此只要求这个图的最小生成树即可。显然，令  $d(u)$  为  $u$  到根的路径异或和，则  $d(u) \oplus d(v)$  就是  $u$  到  $v$  的路径异或和。

剩下的就是 [老原题](#) 了。考虑 Kruskal 过程从小到大连边：将所有  $d(i)$  插入 trie 中，先给子树建好 MST 再往上合并两子树的 MST 合并的过程是要找两个子树中异或的最小值，这个可以暴力在两个子树中递归。由于每个节点只会被暴力搜到  $O(\log V)$  次，因此总复杂度是正确的。如果不放心，也可以启发式合并维护一个子树的值，用另一个子树去查询最小值。

## C - Easy

Upsolved by nikkukun.

## 题目描述

两个正整数序列  $a_1, a_2, \dots, a_k$  和  $b_1, b_2, \dots, b_k$  满足  $\sum_{i=1}^k a_i = n, \sum_{i=1}^k b_i = m$  对于两序列所有可能的情况，求

$$\prod_{i=1}^k \min(a_i, b_i)$$

之和。其中  $1 \leq n, m \leq 10^6, 1 \leq k \leq \min(n, m)$

## 解题思路

首先要将  $\min$  转化为其他方便统计的东西，通常的做法是  $\min(a, b) = \sum_{i=1}^{\infty} [i \leq a] \cdot [i \leq b]$  这样拆，进而变成对额外变量的统计。这个技巧在 [CF 1292C](#) 也有类似应用：

关于一个树上 mex 的推论：

$$\begin{aligned} S &= \sum_{1 \leq u < v \leq n} \mathrm{mex}(u, v) \\ &= \sum_{1 \leq x \leq n} \left( \sum_{\mathrm{mex}(u, v) = x} 1 \right) \\ &= \sum_{1 \leq x \leq n} \left( \sum_{\mathrm{mex}(u, v) \geq x} 1 \right) \\ &= \sum_{1 \leq x \leq n} f(x) \end{aligned}$$

其中  $f(x)$  是满足  $\mathrm{mex}(u, v) \geq x$  的二元组个数。

即是说  $\mathrm{mex}$  和  $\min$  操作都可以用类似方法解决。因此本题有

$$\prod_{i=1}^k \min(a_i, b_i) = \prod_{i=1}^k \sum_{c_i=1}^{\infty} [c_i \leq a_i] \cdot [c_i \leq b_i]$$

后面的和式代表  $i$  位置上  $(a_i, b_i, c_i)$  三元组的合法种类，故由乘法原理可以知道式子实际是对满足条件的  $(\{a_n\}, \{b_n\}, \{c_n\})$  三元组计数。不妨钦定  $n \leq m$  且枚举  $x = \sum_{i=1}^k c_i, x \in [k, n]$  则插板法得到满足条件的  $\{a_n\}$  个数为  $\binom{n-x+k-1}{k-1}$  故可以计算最终答案为

$$\sum_{x=k}^n \binom{n-x+k-1}{k-1} \binom{m-x+k-1}{k-1} \binom{x-1}{k-1}$$

总时间复杂度  $O(T \min(n, m))$

## D - Drop Voicing

Solved by qxforever.

### 题目描述

给一个  $1-n$  的排列  $p$  有两种操作：

1. 把序列最前面数的放到最后面
2. 把序列倒数第二后的数放到最前面

要把原排列变为元排列，求最少的（连续第二种操作）的次数。

### 解题思路

排成一个环之后发现，一操作就是旋转，二操作就是交换，故问题转化成求环排列的最大 LIS 答案即为  $n$  最大 LIS

## E - Bogo Sort

Solved by Potassium.

水题不表。

## F - DPS

Solved by Potassium.

水题不表。

## H - Interval

Upsolved by nikkukun.

### 题目描述

给长度为  $n \leq 10^5$  值域在  $[0, 2^{30})$  的序列  $a_1, a_2, \dots, a_n$  强制在线  $q \leq 10^5$  次询问下标  $[l, r]$  的所有子区间中，区间按位与能获得的值的个数。

### 解题思路

比赛时想到了对于确定的左端点，其右端点最多能产生  $O(\log V)$  个值，因此实际只需要统计  $[l, r]$  里包含了多少个这样的极小区间即可，这是一个二维偏序问题。但是这样显然是不行的，同一个值只能统计一次，当时并没有想到如何消除影响。

题解给出了一种很妙的做法：对同一个值的所有区间，首先消除掉所有包含关系（事实上，如果一开始找区间就选的极小区间，是不会出现除了全等之外的包含关系的），并按左端点依次排序为  $(l_1, r_1), (l_2, r_2), \dots, (l_k, r_k)$  额外添加  $k-1$  个区间  $(l_1, r_2), (l_2, r_3), \dots, (l_{k-1}, r_k)$  并令其贡献为  $-1$ ，则刚才的统计就是对的了。

可以手玩一下上面的构造，当包含了相邻两个区间时会自动让结果  $-1$ ，这个结果在两区间相交与否的情况都成立。

## I - Hard Math Problem

Solved by nikkukun.

水题不表。

## K - Git Merge

Upsolved by nikkukun.

### 题目描述

给一段用 Git Merge 的、有两个 branch 的代码，需要你重新合并出一个最少行的结果，具体操作请参考原题。输入总行数不超过  $4000$  行。

### 解题思路

先把两段源代码还原出来，令  $f(i, j, \text{op})$  为处理了第一段代码的前  $i$  行、第二段代码的前  $j$  行，目前正处于第一段单独区间/第二段单独区间/两段共用区间的最小行数，不同  $\text{op}$  之

间的切换会增加一行的代价[]DP 转移即可。可能需要用哈希判断行相等，用 short 存数组减少空间。

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:  
[https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i\\_dont\\_know\\_png:multi2020-nowcoder-5](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:i_dont_know_png:multi2020-nowcoder-5)

Last update: **2020/08/07 09:24**

