Cycling City

题目大意

给你一张 n 个点 m 条边的无向图，求图中是否存在两个点，满足这两个点之间有三条不共点(不包括起点和终点)的简单路，并输出路径。

数据范围

$1 \leq n, m \leq 2 \cdot 10^5$

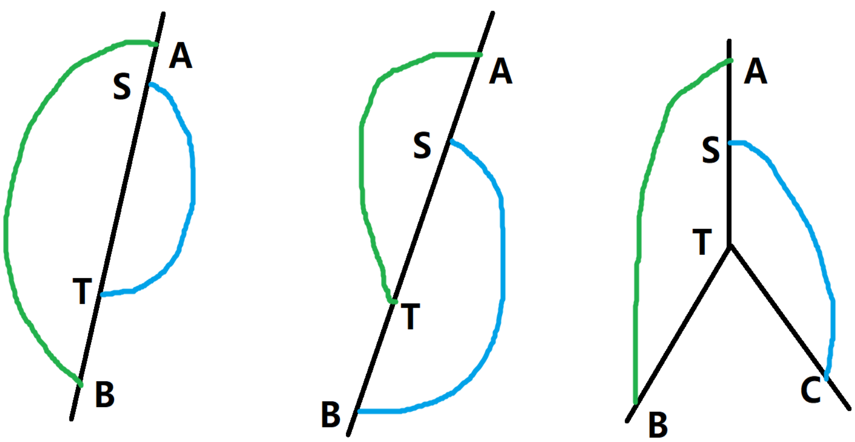
解题过程

我们对原图建出 dfs 树，则有解的充要条件是存在一条树边被至少两条返祖边覆盖(存在两条返祖边覆盖树边的集合有交集)。

证明：

如果不存在这样的边，那么图上两点之间，如果有一条返祖边覆盖了这两个点，就有两条路径；否则，只有一条路径。

如果存在这样的边，则任意找到覆盖他的两条返祖边，他们会有如下三种情况：



1. S-返祖边-T，S-树边-T，S-树边-A-返祖边-B-树边-T
2. S-树边-T，S-树边-A-返祖边-T，S-返祖边-B-树边-T
3. S-树边-T，S-树边-A-返祖边-B-树边-T，S-返祖边-C-树边-T

因此，我们一定能构造出方案。

时间复杂度 $O(n)$。

Permutation and Minimum

题目大意

给你一个长度为 2n 的序列 a，其中一些位置的数不确定，要求将 a 补充成一个 [1,2n] 的排列。定义一个长度为 n 的序列 b，$b_i = min(a_{2i-1}, a_{2i})$，求一共有多少种不同的 b 序列。

数据范围

$1 \le n \le 300$

解题过程

首先，如果 a_{2i-1}, a_{2i} 都已经确定的话，b_i 就是确定的，这时我们不用考虑 a_{2i-1}, a_{2i} 这两个数。

对于剩下的数，考虑从大到小确定每一个值的位置。

如果新加入一个不确定位置的数，那么他有三种可能：

1. 与之前加入的一个不确定位置的数合成一对。

2. 与之前加入的一个确定位置的数合成一对。

3. 准备与之后加入的一个数合成一对。

如果新加入一个确定位置的数，那么他有两种可能：

1. 与之前加入的一个不确定位置的数合成一对。

2. 准备与之后加入的一个数合成一对。

对于准备和之后加入的数匹配的不确定位置的数，他们不影响 b 序列的值，都是等价的。对于那些两个不确定位置的数合成的数对，他们的数量为定值，设为 cnt，我们先不考虑他们的位置，然后最后答案再乘以 $cnt!$。

于是进行 dp，设 $dp[i][j][k]$ 表示从后往前考虑了值 [i,2n]，当前有 j 个确定位置的数准备与之后加入的一个数合成一对，有 k 个不确定位置的数准备与之后加入的数合成一对。

则 dp 的转移为：

1. 数 i 为不确定位置的数，$dp[i][j][k] = dp[i+1][j][k+1] + (j+1) \cdot dp[i+1][j+1][k] + dp[i+1][j][k-1]$。

2. 数 i 为确定位置的数，$dp[i][j][k] = dp[i+1][j][k+1] + dp[i+1][j-1][k]$。

时间复杂度 $O(n^3)$。

Histogram Coloring

题目大意

给一个 10^9 行 n 列的表格，从左向右、从下往上标号，并去掉一些格子，使得第 i 列只有最底下的 h_i 个格子。

要求对剩下的格子用红蓝两种颜色染色，使得所有 $2*2$ 的格子中恰好有 2 个红色格子，2 个蓝色格子，求方案数。

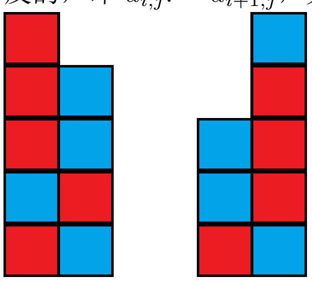
数据范围

$1 \leq n \leq 100, 1 \leq h_i \leq 10^9$

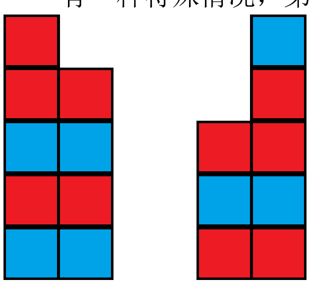
解题过程

定义第 i 列，第 j 行的颜色为 $a_{i,j}$。

在一般情况下，第 i 列与第 i+1 列高度 $\leq min(h_i, h_{i+1})$ 的部分，染色情况恰好是完全相反的，即 $a_{i,j}! = a_{i+1,j}$，如图：



有一种特殊情况，第 i 列上高度 $\leq min(h_i, h_{i+1})$ 的部分红色与蓝色交替出现，如图：



考虑将这个表示在状态中。

因为高度是 10^9 级别的，不能作为状态，但我们只需要知道，对于所有的 i，从底部向上的 h_i 个格子是否为红色与蓝色交替出现。

所以我们将所有的 h_i 排序并去重，生成高度序列 d。求出序列 nh 表示 $h_i = d_{nh_i}$，然后设 $dp[i][j]$ 为考虑前 i 列，第 i 列从下向上数的前 d_j 个格子满足红色与蓝色交替出现，且前 d_{j+1} 个格子不满足。

dp 的初始状态为

$dp[1][0] = 2^{h_1} - 2 \cdot 2^{h_1 - d_1}$

$dp[1][i] = 2 \cdot 2^{(h_1 - d_i)} - 2 \cdot 2^{(h_1 - d_{i+1})}(0 < i < nh_1)$

$dp[1][nh_1] = 2$

dp 的转移为：

1. $h_i \geq h_{i+1}$

$dp[i+1][j] = dp[i][j](0 \leq j < nh_{i+1})$

$dp[i+1][nh_{i+1}] = 2 \cdot \sum_{j=nh_{i+1}}^{nh_i} dp[i][j]$

2. $h_i < h_{i+1}$

$dp[i+1][j] = dp[i][j] \cdot 2^{h_{i+1} - h_i}, (0 \leq j < nh_i)$

$dp[i+1][j] = 2 \cdot dp[i][nh_i] \cdot (2^{h_{i+1} - d_j} - 2^{h_{i+1} - d_{j+1}})(nh_i \leq j < nh_{i+1})$

$dp[i+1][nh_{i+1}] = 2 \cdot dp[i][nh_i]$

最后 $ans = \sum_{i=0}^{nh_n} dp[n][i]$

复杂度 $O(n^2)$