

杂题选讲

By RymCold,
From zpk Fan Club.

就像 每一场考试**都有**一道签到题一样
一次交流也要从一道**简单**题 开始。

CF276D Little Girl and Maximum XOR

[Link:](#)

给出两个整数 l, r , 求 $\max\{a \oplus b\}$, 其中 $l \leq a \leq b \leq r$;

$$1 \leq l \leq r \leq 10^{18}$$

Sample:

输入 #2

复制

输出 #2

复制

114514

1919810

2097151

这题难度居然 1700 。

我们从高到低找到 l 和 r 的最高的不一样的一位 k 。

那么答案就是 $2^{k+1} - 1$ 。

感性理解一下挺对的；

理性分析一下就是贪心。

不理解的话可以考虑一个极限情况

100010111 0 111111111111

100010111 1 000000000000

CF1415D Xor-Gun

Link:

给定一个长为 n 的不降序列，每次操作可以任选相邻的两个数 a_i, a_{i+1} ，并将这两个数替换为两个数按位异或的结果，现在需要破坏序列的不降，求最少操作次数，无解输出 -1;

$$2 \leq n \leq 1e5, 1 \leq a_i \leq 1e9;$$

CF 2000 的题。

首先有一个比较显然的性质，

如果有三个连续的数二进制最高位相同，操作次数为 1。

那 要是不存在这样的三个数，也就是说对于每个二进制位，最多只会有两个数满足最高位一样。

观察 a 的数据范围，发现需要枚举的数只剩下 $2 \times 30 = 60$,

$n > 60$ 一定有解为 1，这部分遍历的时候就求出来啦。

所以就可以 (n^3) 辣；

CF1375E Inversion SwapSort

[Link:](#)

给定一个长度为 n 的序列 a , 构造一个由 a 中所有逆序对 $(i_1, j_1), (i_2, j_2) \cdots (i_m, j_m)$ 构成的一个排列 p ,

使得顺次交换 $(a_{i_1}, a_{j_1}), (a_{i_2}, a_{j_2}) \cdots (a_{i_m}, a_{j_m})$ 后序列单调不降。

$1 \leq n \leq 1e3, 1 \leq a_i \leq 1e9;$

2500 中比较简单的构造。

如果 a 是一个排列的话是很简单的,直接从后向前从大往小放就好了;

不是一个排列的话其实没有什么区别。我们从小到大考虑,以权值为第一关键字,以下标为第二关键字排序,对于当前数据 i ,只要找到所有的 j 满足 $a_j < a_i$,然后依此交换就好了;

发现这个东西其实就是个冒泡+选择排序的过程。

形式化一点,就是顺次交换 $[pos_{num_{n+1}}, n], [pos_{num_{n+2}}, n] \cdots [pos_{num_n}, n]$;

证明正确性:每次交换使得排名为 i 的数字正好排在了第 i 位,并且保证了其他各位的大小**相对**关系不变;

P4745 [CERC2017]Gambling Guide

Link:

一个铁路系统由 n 个城市和 m 条双向铁路组成。铁路票只能在安装在每个城市的自动售票机购买。不幸的是，黑客们已经篡改了这些售票机，现在它们有下面的规则：

当 a 市的售票机有一个硬币投入时，机器会发一张从 a 市到随机一个邻市的单程票。

你需要从城市 1 到城市 n 。你知道机器是怎么工作的并且有一份铁路系统的地图。在每一个城市，当你买了一张票时，你可以选择立即使用它后到达目的地，或者是丢掉它并买一张新票。你可以无限制的购买的票。当你到达城市 n ，旅行就会结束。

你需要确定一个满足以下条件的策略：

- 旅行最终到达终点的概率为 1。
- 花在旅行上的硬币的期望值越少越好。

输出这个期望值。

$1 \leq n \leq m \leq 300000$;

时间限制 : $2s$;

空间限制 : $500MB$;

我做这个题的时候比较草率，
推式子的时候最重要的一步和转移的时候最重要的一步直接被我忽略了；然后当场讲这个题的时候效果直接爆炸
本当にごめんなさい；
特别鸣谢 AωЯ2020 くん；

考虑DP;

一开始很懵, 因为啥也不知道;

看这个数据范围发现不能设二维的状态, 那么尝试从几个一维状态入手;

我们可以设一个 f_i 表示从某个点出发到终点的期望? 亦或是到某个点的期望?

但其实这两个方案推式子的时候都非常困难;

考虑无论正推还是逆推, 初始状态都是 $f_n = 0$;

很自然觉得应该逆推。然后设 f_i 表示到某个点的期望步数。虽然没有正确性证明, 但毕竟我们现在一点方向都没有, 至少还是要尝试往一个方向走试一试的;

那么首先尝试列一下式子。有：

$$f_i = 1 + \frac{1}{\deg_i} \sum_{i,j \in E} \min(f_i, f_j)$$

虽然这个里面我们除了 \deg_i 以外什么也不知道，但是我们除了这个式子以外也确实啥也不知道了；

解释一下这个 \min 的含义：如果 $f_j > f_i$ 的话，转移过来还是 f_i ，因为还不如待在原地不动而不是选择一个更劣的；

然后发现不会了。这个东西巨大多麻烦。

不过至少可以确认的一点是肯定不用这个东西转移；要么硬求 f 要么拆式子；

考虑到我们要求解的是 f ，那么就只能尝试拆一下式子了；

不难发现当前这个点只能被从 f 已经确定的点转移过来，但是我们并不知道是哪几个。考虑到这对原式简化是有帮助的，那么我们就设一下嘛。设一个 c_i 表示我们当前处理到 i 这个点时已经确定的与 i 直接相连的点数。则上式可以进一步转化：

$$f_i = 1 + (1 - \frac{c_i}{\deg_i})f_i + \frac{1}{\deg_i} \sum_{i,j \in E \& f_j < f_i} f_j$$

于是

$$f_i = \frac{\deg_i + \sum_{i,j \in E \& f_j < f_i} f_j}{c_i}$$

感动。我们基本上全拆开了。

看似无厘头的问题好像就要被我们求出解啦。

接下来只要解决 c 的求法以及转移的方式就做完了。

观察式子，不难发现我们转移到的点 j 一定满足 $f_j > f_i$ ；

于是不难想到较小的点只会被转移一次，这意味着所有点只会被转移一次。

这个 c_i 就可以很自然的在转移过程中求出了。

发现上面这个形式很熟悉。

我们每次选择一个 f 值较小的点，然后向 f 大的点转移；

并且保证每个点只会被遍历一次；

这不就是 Dijkstra 吗，直接拉过来转移就好了。

时间复杂度 $O(m \log n)$ ；

Code:

Thank you

诚邀您加入 zpk Fan Club! ([AT/CF](#))