

## 目录

A、 Small change.....	1
B、 Scoop water.....	1
C、 CX's dream.....	1
D、 CX and girls.....	2
E、 ZZY's new company.....	2
F、 ZZY and his little friends .....	3
G、 Apple tree.....	3
H、 A hard-problem .....	3
I、 The contest .....	4
J、 Scholarship .....	4

## A、Small change

签到题，要表示出 $(1,N)$ 的所有数，至少需要  $N$  的二进制位多个数。

如 $(5)_{10}=(101)_2$  所以需要一张 1 元的，一张 2 元的，一张 4 元的。

## B、Scoop water

首先水瓢 A 和水瓢 B 都要使用  $N$  次，我们记  $dp[i]$  表示要使用  $i$  次 A 和 B 时的方法数。

在舀水过程中，如果第一次出现使用水瓢 B 之后水缸正好空了，我们姑且记这是第  $i$  次使用水瓢 B，那么这次舀的水是第一次使用水瓢 A 舀进去的。第一次使用水瓢 A 和第  $i$  次使用水瓢 B 之间的

过程则是使用了  $(i-1)$  次水瓢 A 和  $(i-1)$  次水瓢 B，方法数为  $dp[i-1]$ ；而第  $i$  次使用水瓢 B 之后的过程也类似，为  $dp[N-1-(i-1)]$ ，所以  $dp[N] = \text{SUM}(dp[i]*dp[N-1-i])$  为我们所要求的答案。

## C、CX's dream

CX 的每个目标达成的必要条件是完成了对应的付出，也就是说某些付出是某个目标的必要条件，而且每个付出又是不存在任何必要条件的。由此可以想到最大权闭合子图的模型。

每个目标和每个付出都看成有向图中的点，对于每个目标  $A_i$ ，超级源点向其做边容量为其达到所能获得的权值，并且  $A_i$  向其所需的付出做边，容量为无穷大。对于每个付出  $B_i$ ，所起完成能获得的权值为负，超级源点往其做边，容量为其权值，若  $B_i$  的权值为负，则其向超级汇点做边，容量为其权值的相反数。

把所有的正权值加起来记为  $\text{sum}$ ,  $\text{sum}$  减去超级源点到超级汇点的最大流就是答案。

但这样只是求出了能获得的最大权值,而本题还要在所能获得的权值的权值最大的情况下让完成的目标最多,这就需要设置偏移量来维护。由于梦想最多 1000 个,所以我设置偏移量为 2000,那么该图中所有的边容量乘以 2000,在超级源点往目标点做边时,容量加一。如此处理,得到最大流后就可以分解出最大点权已经满足最大点权下完成的最大目标数。

## D、CX and girls

简单最短路,几种方法:

1、直接最短路,更新的时候两个条件带着跑,当路径更短时路径和学妹数同时更新,当路径长度相同而学妹数量更多时更新学妹数量。

2、拆点,用偏移量维护两个优先级不同的最优值。每个点都拆成两个点,两点间的距离为其学妹数量的相反数,而对于每个点之间的边则要乘以偏移量,因为总学妹数量小于等于 50000,

所以我设置偏移量为 80000,这样跑出最短路后就可以分解出最短路径长度以及在满足路劲最短的情况下看到的最多学妹。

## E、ZZY's new company

这题的大致思路就是把公司成员全部编号,然后用线段树来维护,关键的是编号方式。

我使用的是从跟开始广度优先遍历的顺序来编号,那么某个人的第  $i$  层的自结点编号就肯定是连续的了。我们可以使用倍增法来在  $\log(N)$  的时间内,找出某个节点的第  $i$  层子所有节点中的编号

最小的孩子和编号最大的孩子，这样就可以用线段树来维护这段连续的虚列了。

具体方法是用  $chl[i][j]$  表示  $i$  节点一直往最左边孩子走时的第  $2^j$  层的孩子，同样  $chr[i][j]$  表示一直往右走时的第  $2^j$  层的孩子。

详情可以 [baidu](#) 用倍增法查找最近公共祖先。

## F、ZZY and his little friends

这题就是求出所有数字对的异或值中的最大值是否超过  $M$ 。

我们在求  $i$  和  $i$  前面的所有数中的最大值时可以把前  $(i-1)$  个数用字典树(二进制序列串,从高位开始插)表示出来，然后用第  $i$  个数的二进制序列串从根开始遍历这棵字典树，遍历时要尽量往相反的方向走，如这个数的该位置为 0 时如果树的节点有右孩子(表示 1)则一定往右边走，遍历完之后就把这个数也插入字典树中。

## G、Apple tree

国家集训队论文(徐持衡,树形依赖背包,泛化物品等)

## H、A hard-problem

首先对于倍数问题答案直接是:  $(A/p)*(B/p)$

剩下的是一个典型的统计 gcd 的问题。用的是 mobius 函数的性质。

首先，在  $10^7$  中的任何一个数不会有很多的约数，最多也就 500 左右，我们可以用类似素数筛选的方法筛选得到每个数的函数值(这里不能枚举每个数求函数值，会超时)，然后用数组  $f[x]$  表示前  $x$  项 mobius 函数值得和，求解的时候分块处理就好了。

总的时间复杂度为:

预处理:  $n \cdot \log(n)$  实际效果比理论要好很多

最后询问:  $q \cdot (\sqrt{A} + \sqrt{B})$  (显然不会超时)

整个题目在求解时候用 `int` 写比较快, 但是在算 `ans` 的时候要先转为 `long long`。

题目的时间限制和空间限制都是 2+倍, 有点卡时间, 注意不要把常数写大了。

(好吧, 我承认在这个题目上我没什么节操)

## I、The contest

并查集+分组背包

## J、Scholarship

水题一发。

首先说明这个题目应该是有贪心的解法。

我这里说说我最初出题的目的是用数位 DP+二分求解的。

预处理数位 dp 所需要的函数, 然后从  $x+1$  到 100000000 一步步二分就可以咯。

注意一个坑, 那就是如果  $x$  为负数直接输出 0。