大厂出题源：codeforce

# 腾讯3-21实习生招聘

1. 给了一颗树，节点最多1000个，1000次查询，问根到这个数字x都经历了哪些点

每次查询返回的是个Listnode的指针

树，链表

2.一个数字n，三种操作 n=n-1, n=n/2(n是偶数)， n=n/3(n是三的倍数)

返回最小操作步数

n（1-10^9）数字大 dp装不下

递归分治或者迭代都行

带记忆的dfs

3.那个数组，然后查询q次，每次查询给了若干数组的索引，求合并数组的第k小

4. n个员工，每个员工有对奖金有上下限，奖金w。求，怎么分配使得员工奖金中位数最高

5. n件商品，分别对应商品的价格。在满足总金额是m倍数的情况下，购买金额最高

0-1背包问题，动态规划

# 阿里3-15实习生招聘

5． 一根长度为 n 的蜡烛，随机分成两段（一共有 n-1 种分割方式，假设概率均等），将两段蜡烛同时燃烧，如果一根燃尽，另一根还剩长度 >= 2，则把剩余这根蜡烛继续切分，然后分别点燃，直到蜡烛燃尽。注意第二次切分之后不再继续切分。

求蜡烛燃尽的期望时间。

长度为n蜡烛切一次花费的时间期望值是：

数学题:

长度为n；

切一次：i，n-i

t1=min(i,n-i);

//剩余蜡烛left=abs(n-2\*i)

返回可能性和时间总和

第二次切是left-j,j

时间max(left-j,j)

该可能性对应时间：max(abs(n-2\*i)-j,j)+min(I,n-i)

两个循环可以遍历出所有可能性，和对应时间

便可以求均值了

# 阿里3月17-3月19

6. n副扑克，张数为m，大小为1～m，每幅扑克抽一张，求和恰好为k的组合数，结果对10e9+7取余数。  
思路：[动态规划](/jump/super-jump/word?word=%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92)。和为i，j副扑克，，dp[i][j] = dp[i-1][j-1]+...+dp[i-m][j-1](此处需要判断 i-m>0 )。初始化，j=1，i<=m,dp[i][j] = 1; i==j,dp[i][j] = 1,后面的情况不可能存在，所以推出循环。

一共取n次，每次都可以是牌种类0-m张

完全背包问题背包问题

Dp[i][j]:取i次和为j组合数排列问题

Dp[i][j]=dp[i-1][j-nums[k]]

三重循环

7. n个数，分成两份，舍去总数大的，剩下的是本轮得分；直到最后只剩下一个数字，请问最大得分是多少？

# 网易3-27

1. 曼哈顿距离最远的所有点
2. 三次方回文数

# 美团3-27

1. 长度为n的字符串；可替换任意一个数

回文优先，数字最小优先

使得输入的字符串修改一个最美程度最高

1. 两个技能需要消耗对应mp值，使用技能就会赢；

不适用可能赢可能输

小美被连续打败三次就输掉游戏

10 7 3 回合数，技能消耗mp

FTFFFTFFFF 表示不使用技能输赢

求最少需要多多mp才能赢

1. N个数，询问m次。每次询问x数第一次出现位置和最后一次出现位置；

没有出现过回答0

6 4

2 3 1 2 3 3

1

2

3

4

输出

3 3

1 4

1. N个数的数组，最多从中挑选k个连续的数。

对这些数做异或操作。求最大结果

3 2

1 2 4

# 百度0330

1. n个人，对应期待值ai， 现有m个期待值，bi，求满足每个人期待值的个数。

2. 求交集问题。m个特征，每个特征K个区间。求满足所有特征的区间

3. n个人，m个食堂，每个食堂有num[i]个窗口。也就是说每个食堂窗口人数均分。

每个人去每个食堂的概率相同。求最长队伍的期望值

Cout的数据又会被cin进来？？

**√网易互娱0331**

1. √本金100000，先涨m天后跌n天，每天涨的金额是相同的x，每天跌的金额也是相同的y。求这个基金到达P元至少需要多少天？如果无法实现返回-1

每个周期总共盈利：m\*x-n\*y; 每个周期中，最多涨n\*x

1（M） 1(N) 4(X) 2(Y) 100005(P)s

1. Printf结构的判断，每个格式，输入会进入几次if判断
2. √变量替换，查找受影响的句子

4

A wo

B A hao

C A B dc

D C skd

//记录直接相关联的，找到所有相关联的

记录变量和直接包含的变量

记录常量和包含该常量的句子数目。

//变量修改用层序遍历+visited防止重复

# 腾讯0418

1. √寻找链表所有逆序中的最小字典序

4->23->2 返回 2->4->23

2-2 -5-2-2 返回2 -2-2-2-5

1. 发广告，n个用户，k个广告

第i用户要求每t[i]时间可发送广告

如果某一时刻可以发给多个用户，则优先发给编号小的用户。打印出按时间顺序发给的用户编号

如：

3 3

3（用户1）

6 （用户2）

9 （用户3）

第一个发给用户1，时刻6的时可以发给用户1或2，发给1；第三封发给2

K 10^6 100万

用优先队列超时了 只过70%

1. N个游戏，第i个游戏需要在t[i]前完成，如果没完成则扣除积分w[i],求最少扣除的积分数。

3

3 1 1

3 6 9

这道题可能是理解不到位，写了但是没过

1. 比较字符串s1和s2,如果长度为奇数就正常比较(字符串中只存在a,b)，如果是偶数，可以将a分成a1,a2, b可以分成b1,b2,若a1=b1&&a2=b2或者a1=b2&&a2=b1（递归）则相同。
2. 打地鼠，mp[i][j]表示每隔mp[i][j]秒在I,j位置就有地鼠出现。整个土地为m\*n.

在0时刻，小明站在(1,1)位置，每一秒可以上下左右走一格，可以a到b，下一秒就不能b->a。

最后要回到终点，否则只打0只地鼠

2 2 6

1 1

1 1

最多能打6只

# 题目1

给定两个数 n 和 m，对 n 中的数重新排列，计算出经过重新排列后所得到的数中满足不含有前导零并且能够整除 m 的数字有多少个？  
例如，n = 520，那么重新排列后：520，502，250，205，025，052，能被 m = 2整除的，并且无前导零的为 520，502，250三个

# 题目2

包粽子, 四个数n, m, c0, d0, 一共n 克面粉， m种馅料

然后m行,每行四个数ai, bi, ci, di, ai 表示一共多少克该种馅料

每个粽子包法, bi克第i种馅料 + ci 克面粉, 收益di, 或者 c0 克面粉, 不包馅料, 收益d0

求最大收益

# 题目3

给定一个数x，数据对 (a, b)使得a ^ b ^ x能达到最大，求使|a - b|最小的方案总数有多少。

x,a,b的范围都是0 - （2^31 次方-1）

0 -> 2, 100 -> 16

# 题目4

在漆黑的夜里，N位旅行者来到了一座狭窄而且没有护栏的桥边。如果不借助手电筒的话，大家是无论如何也不敢过桥去的。不幸的是，N个人一共只带了一只手电筒，而桥窄得只够让两个人同时过。如果各自单独过桥的话，N人所需要的时间已知；而如果两人同时过桥，所需要的时间就是走得比较慢的那个人单独行动时所需的时间。问题是，如何设计一个方案，让这N人尽快过桥。假设N <= 1000

https://blog.csdn.net/wuzhekai1985/article/details/6846934

# 题目5

https://www.nowcoder.com/discuss/461399?type=post&order=time&pos=&page=1&channel=-1&source\_id=search\_post\_nctrack&subType=5

有个藏宝架有n层，每层的宝物数量不一，每个宝物都有其价值，现在要求拿出m个宝物，并且需要遵守规则**：**

1. 每次只能拿选定层的两端的宝物
2. 要拿出的m个宝物的总价值是各种方案里最大的

输入： n m  
下面每行代表每层，且第一个数是这层宝物的数量k，后面的则是k个宝物的价值  
4 1 2 4 5  
5 1 2 4 5 5  
样例：  
2 3  
2 3 2  
4 1 4 1 5  
输出：5+3+2=10

# 题目6

街区最短路径问题（网上可以搜索到）

# 题目7

**leetcode205**

给出一个字符串 a = "abcaa"(任给)，和它对应对编码后对字符串 b="skbss",判断该编码是否合法。

# 题目8

小强战斗怪兽，n

题目9

题目10

行测题

<https://www.nowcoder.com/discuss/520435?type=post&order=time&pos=&page=1&channel=-1&source_id=search_post_nctrack&subType=5>

# 阿里2021秋招笔试（8.28）——研发开发岗

阿里笔试比字节还难，如果你能 A 0-1道，那么你是个努力的臭弟弟，如果你能 A 一道，人上人，能 A 两道，你一定是Acmer。  
祝所有秋招的同学offer拿到手软

# 一、字符串翻转

对于一个01字符串，每次只能

1. 交换任意两个元素
2. 把一个0变成1或者把一个1变成0
3. 翻转整个串

## 代码：A了0.7

import java.util.Scanner;

/\*\*

\* Created by IntelliJ IDEA.

\*

\* @Author: 张志浩 Zhang Zhihao

\* @Email: 3382885270@qq.com

\* @Date: 2020/8/28

\* @Time: 19:13

\* @Version: 1.0

\* @Description: Description

\*/

public class First {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

String s = sc.next();

String t = sc.next();

sc.close();

int a1 = 0, a2 = 0, t1 = 0, t2 = 0;

for (int i = 0; i < n; i++) {

char p1 = s.charAt(i), p2 = t.charAt(i);

if (p1 != p2) {

if (p1 != '0') t1++;

else t2++;

}

}

int tmp = 0;

if (t1 < t2) {

tmp = t1;

t1 = t2;

t2 = tmp;

}

a1 = t1;

t1 = 0;

t2 = 0;

for (int i = 0; i < n; i++) {

char p1 = s.charAt(i), p2 = t.charAt(n - i - 1);

if (p1 != p2) {

if (p1 == '0') t1++;

else t2++;

}

}

if (t1 < t2) {

tmp = t1;

t1 = t2;

t2 = tmp;

}

a2 = t1 + t2;

System.out.println(Math.min(a1, a2));

}

}

* 1
* 2
* 3
* 4
* 5
* 6
* 7
* 8
* 9
* 10
* 11
* 12
* 13
* 14
* 15
* 16
* 17
* 18
* 19
* 20
* 21
* 22
* 23
* 24
* 25
* 26
* 27
* 28
* 29
* 30
* 31
* 32
* 33
* 34
* 35
* 36
* 37
* 38
* 39
* 40
* 41
* 42
* 43
* 44
* 45
* 46
* 47
* 48
* 49
* 50
* 51
* 52
* 53
* 54
* 55

# 二、数字排列

给定两个数 n 和 m，对 n 中的数重新排列，计算出经过重新排列后所得到的数中满足不含有前导零并且能够整除 m 的数字有多少个？  
例如，n = 520，那么重新排列后：520，502，250，205，025，052，能被 m = 2整除的，并且无前导零的为 520，502，250三个

## 代码：全A

import java.util.Scanner;

/\*\*

\* Created by IntelliJ IDEA.

\*

\* @Author: 张志浩 Zhang Zhihao

\* @Email: 3382885270@qq.com

\* @Date: 2020/8/28

\* @Time: 19:42

\* @Version: 1.0

\* @Description: Description

\*/

public class Second {

private static long[][] dp;

private static char[] s;

private static int m;

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

s = ("" + sc.nextLong()).toCharArray();

m = sc.nextInt();

sc.close();

dp = new long[(int) Math.pow(2, s.length)][m];

for (long[] a : dp)

for (int i = 0; i < a.length; i++)

a[i] = -1;

System.out.println(getDP(0, 0));

}

private static long getDP(int mask, int mod) {

if (dp[mask][mod] >= 0)

return dp[mask][mod];

if (mask == dp.length - 1) {

dp[mask][mod] = mod == 0 ? 1 : 0;

return dp[mask][mod];

}

dp[mask][mod] = 0;

boolean[] used = new boolean[10];

for (int i = 0; i < s.length; i++)

if (!used[s[i] - '0'] && (mask | 1 << i) != mask && (mask > 0 || s[i] != '0')) {

dp[mask][mod] += getDP(mask | 1 << i, (10 \* mod + (s[i] - '0')) % m);

used[s[i] - '0'] = true;

}

return dp[mask][mod];

}

}