

A题 大海景房

题意

从左往右找到前面没有比本身大的那些数。

算法

这道题有同学想到了最长不下降子序列？可行吗？

我们是要看海，最西边的那座山最高，后面的最长不下降子序列再长都没有用的。如：10 1 2 3
4 5 6 7

但是，一看数据范围，这么小，暴力枚举不是更直接吗？从左往右查找每个数，把前面的最大高度记下来，后面的数 \geq 这个高度就 $\text{ans}++$ 。

B题 人见人爱的食物

题意

求 A_{ij} 中，出现 N 次的数

算法

数据范围这么小，用桶数组 $a[i]$ ，表示 i 出现了几次。输出出现 N 次的 i 就行了。

C题 涂瓷砖

题意

把字符串 S 变成 $01010101\dots$ 或者 $10101010\dots$ 至少需要改变几个颜色。

算法

就两种情况，分别讨论一下就好了，以1开头的话，需要改变几个颜色，以0开头的话，需要改变几个？取最小。

D题 怪兽打架

题意

求这些数两两相减后，剩的最小数。

算法

怪物的血量一直两两相减,类似于辗转相减法。

可以证明,最后存活怪物血量最小值即为所有怪物初始血量的最大公约数gcd.

E题 火柴数字

题意

已知 N 根火柴和 A 序列中这几个数，拼出一个最大整数。

算法

动态规划DP。

阶段：以每个数字划分阶段讨论。

状态：状态即子问题。总问题是 N 根火柴能拼成的最大整数。那子问题就是 i 根火柴能拼成的最大整数。dp[i]表示i根柴能拼成的最大整数。

分析

以样例1为例 来分析。

20 4

3 7 8 4

数组A[] = {8, 7, 4, 3}, 对应火柴数量的数组map[] = {7, 3, 4, 5}

为了很方便地将数字连接起来。把数字当成字符串来处理比较方便, string类型, 可以直接使用
"+" 连接。比如" 7" + " 7" = " 77" ,或" 77777" + " 3" = "777773" 。

string dp[100005]

样例1中 $N = 20$, 所以输出dp[20]就行了。

根据样例1中数据的对应关系, dp[7]=" 8" , dp[3]=" 7" , dp[4] = "4" , dp[5] = "3" 。

接着从1开始枚举dp。

N = 1时, dp[1] = "" , 也就是说, {8, 7, 4, 3} 无法组成各位数之和为1的数。

N = 2时, dp[2] = "" , 也就是说, {8, 7, 4, 3} 无法组成各位数之和为2的数。

N = 3时, dp[3] = "7" 。

N = 4时, dp[4] = "4" 。

N = 5时, dp[5] = "3" 。

N = 6时, $6 = 3 + 3$ 。3对应的数是7, 所以dp[6] = "7" + "7" = "77" 。

N = 7时, dp[7] = "7" 。

另外 $7 = 3 + 4 = 4 + 3$, 3对应的是7, 4对应的是4。所以dp[7] =
"74" 或dp[7] = "47" 。

取最大值dp[7] = "74" 。

N = 8时, $dp[8] = 3 + 5 = 4 + 4 = 5 + 3$ 。3对应着7, 4对应着4, 5对应着3, 则dp[8] =
"73" 或 "44" 或 "37" , 取最大值dp[8] = "73" 。

N = 9时, dp[9] = dp[6] + 3 = "777" 。

.....

最终, dp[20]即为所求。

F题 异或值与和

题意

求满足题目条件的数据有多少对。

算法

此题，首先要知道：

要满足 $a+b = a \oplus b$ 。那么a和b的对应二进制位上不能同时出现1（自己结合样例验证一下），所以对于a,b的每一位对应关系只能是（0，0）或者（0，1）或者（1，0）这三种情况。

因为 $a+b \leq L$ ，所以此题我们要枚举 $\leq L$ 的所有数来讨论，但我们不用从1遍历到L（会超时）。而是按照数位来划分阶段，从左往右讨论。

状态 $f[i][0]$ 表示到第i位为止，满足条件的a+b的和以0结尾的个数。

状态 $f[i][1]$ 表示到第i位为止，满足条件的a+b的和以1结尾的个数。

$f[i][0]$ 可以由前一个状态转移过来，前一个状态是什么？是不是 $f[i-1][0]$ 和 $f[i-1][1]$ ？

如果当前位为1：

$f[i][0] = 3 * f[i-1][0] + f[i-1][1];$

$f[i][1] = 2 * f[i-1][1];$

如果当前位为0：

$f[i][0] = 3 * f[i-1][0];$

$f[i][1] = 2 * f[i-1][1];$

考点

数位DP，还没学过的同学，可以搜索“数位DP”自学一下。

数位DP是一种计数用的DP，一般就是要统计一个区间【L,R】内满足一些条件数的个数。所谓数位DP，字面意思就是在数位上进行DP。数位的含义：一个数有个位、十位、百位、千位.....数的每一位就是数位！不过，这道题数位是各二进制位。