实验舱
青少年编程
走近科学 走进名校

提高算法班

数位DP、状态压缩DP

Mas

数位DP

最朴素的计数是从小到大开始依次加一

对于位数比较多的数，这样的过程中有许多重复的部分

如从 7000 ~ 7999、从 8000 ~ 8999、从 9000 ~ 9999 的过程非常相似

后三位从 000 ~ 999，不同之处在于千位这一位

一些场景下可将重复部分的贡献使用数组记录

根据题目具体要求设置状态，用递推或 DP 的方式进行状态转移

可记忆化搜索或递推统计答案

为了不重不漏地统计不超过上限的答案，往往 **从高到低** 讨论每一位

数位 DP 中通常会利用常规计数问题技巧（将一个区间内的答案拆成两部分相减）

$$\mathrm{ans}_{L\sim R} = \mathrm{ans}_{0\sim R} - \mathrm{ans}_{0\sim L-1}$$

#752、不要62

题目描述

杭州人称那些傻乎乎粘嗒嗒的人为 62 (音: laoer)

杭州交管局经常会扩充一些的士车牌照

新近出来一个好消息,以后上牌照,不再含有不吉利的数字了,这样一来,就可以消除个别的士司机和乘客的心理障碍,更安全地服务大众

不吉利的数字为所有含有 4 或 62 的号码

例如: $62315, 73418, 88914$ 都属于不吉利号码

但是, 61152 虽然含有 6 和 2 ,但不是 62 连号,所以不属于不吉利数字之列

你的任务是,对于每次给出的一个牌照区间号,推断出交管局今后又要实际上给多少辆新的士车上牌照了

输入格式

输入的都是整数对 n, m

如果遇到都是 0 的整数对 , 则输入结束

输出格式

对于每个整数对,输出一个不含有不吉利数字的统计个数

该数值占一行位置

数据范围

对于全部数据 $0 < n \le m < 10^9$,保证测试数据不超过 10000 组

#752、不要62

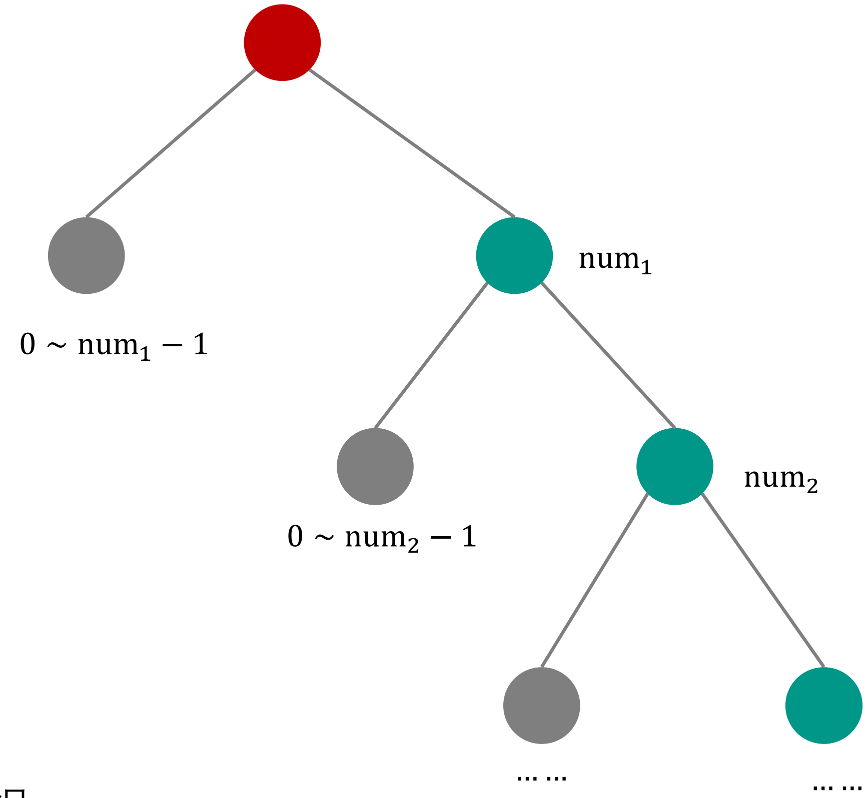
思路1

设 $\mathrm{dp}[i][j]$ 表示有 i 个数位最高位为 j 的合法数字个数

当 $j=4$ 时有 $\mathrm{dp}[i][j]=0$

当 $j\neq 6 \wedge k\neq 2$ 时

$$\mathrm{dp}[i][j]=\sum_{k=0}^{9}\mathrm{dp}[i-1][k]$$



对于一个数 X 将其各数位从高到低拆入 num 数组，从高到低位考虑

记当前为从低到高第 i 个数位，前一数位记为 pre，对于每个数位有两类情况

- 当前数位放 $0\sim \mathrm{num}_i-1$

  排除当前数位 2 且 $\mathrm{pre}=6$ 时情况，累加 $\sum_{j=0}^{\mathrm{num}_i-1}\mathrm{dp}[i][j]$

#752、不要62

- 第 i 个数位放 num_i

  若 $\mathrm{pre}=6$ 且当前数位为 2 或 当前数位为 4 ，结束累加

若到了最低位时依然合法说明原数也是合法的，方案数加一

思路2

设 $\mathrm{dp}[i][0]$ 为 i 个数位的 合法 数量，$\mathrm{dp}[i][1]$ 为 i 个数位以 2 开头的 **合法** 数量，$\mathrm{dp}[i][2]$ 为 i 个数位 **不合法** 的数量

考虑首位追加数字

- 对于合法的方案

$$\mathrm{dp}[i][0]=9\times\mathrm{dp}[i-1][0]-\mathrm{dp}[i-1][1]$$

  其中 $9\times\mathrm{dp}[i-1][0]$ 为当前位不追加 4 的方案数，$\mathrm{dp}[i-1][1]$ 为当前位追加 6 的方案数

- 对于以 2 开头的方案

$$\mathrm{dp}[i][1]=\mathrm{dp}[i-1][0]$$

#752、不要62

- 对于不合法的方案

$$dp[i][2] = 10 \times dp[i-1][2] + dp[i-1][1] + dp[i-1][0]$$

其中 $10 \times dp[i-1][2]$ 为任意追加数位，$dp[i-1][1]$ 为首位追加 6 的方案数

$dp[i-1][0]$ 为首位追加 4 的方案数

对于一个区间 $[1, X]$，只需用 X 减去统计出非法数字数量 cnt

考虑求出 cnt

将数字从 低到高逆序 拆分存入数组中

以 $X = 5863627$ 为例 $num[1 \sim 7] = \{7,2,6,3,6,8,5\}$

首先

$$cnt \longleftarrow cnt + num_i \times dp[i-1][2]$$

以 $num_7 = 5$ 为例累加 0??????,1??????,2??????,3??????,4?????? 的方案

若 $num_i > 4$

#752、不要62

$$\text{cnt} \leftarrow \text{cnt} + \text{dp}[i-1][0]$$

以 $\text{num}_7 = 5$ 为例累加 $4??????$ 的方案

若 $\text{num}_i > 6$

$$\text{cnt} \leftarrow \text{cnt} + \text{dp}[i-1][1]$$

以 $\text{num}_6 = 8$ 为例累加 $62????$ 的非法数量

其次若 $\text{num}_{i+1} = 6 \wedge \text{num}_i > 2$

$$\text{cnt} \leftarrow \text{cnt} + \text{dp}[i][1]$$

以 $\text{num}_5 = 6, \text{num}_4 = 3$ 为例累加 $62???$ 的非法数量

若 $\text{num}_i = 4$ 或 $\text{num}_{i+1} = 6$ 且 $\text{num}_i = 2$ ，后续所有的数位的合法方案都将变为非法

$$\text{cnt} \leftarrow \text{cnt} + \text{num}_i \times \text{dp}[i-1][0]$$

如 $\text{num}_3 = 6, \text{num}_2 = 2$ 对于 $\text{num}_1 = 7$

#752、不要62

前面已出现了不合法情况需要加上 7

由于最后一个数位并未考虑 X 自身

根据 X + 1 进行计算即可

思路3

将数字从高至低拆至数组

每层枚举一个数

若高位已经枚举到其上界

那么当前层的上界则为 num_i，否则该层的上界为 9

跳过 4 以及前一位为 6 当前位为 2 的情况

记忆化搜索即可

三种思路时间复杂度都接近 $O(\lg n)$，记忆化搜索的方式常数稍大

```
int dfs(int len, int pre, bool limit)
{
    if (!len)
        return 1;
    if (~dp[len][pre][limit])
        return dp[len][pre][limit];
    int res = 0;
    for (int i = 0; i <= (limit ? num[len] : 9); i++)
    {
        if (pre == 6 && i == 2 || i == 4)
            continue;
        res += dfs(len - 1, i, limit && i == num[len]);
    }
    return dp[len][pre][limit] = res;
}
int cal(int x)
{
    pos = 0, memset(dp, -1, sizeof dp);
    while (x)
        num[++pos] = x % 10, x /= 10;
    return dfs(pos, -1, true);
}
```

#2869、数字游戏

题目描述

科协里最近很流行数字游戏

某人命名了一种不降数，这种数字必须满足从左到右各位数字成小于等于的关系

如 $123, 446$

现在大家决定玩一个游戏

指定一个整数闭区间 $[a, b]$ ，问这个区间内有多少个不降数

输入格式

有多组测试数据

每组只含两个数字 a, b ,意义如题目描述

输出格式

每行给出一个测试数据的答案,即 $[a, b]$ 之间有多少不降数

数据范围

对于全部数据 $1 \le a \le b \le 2^{31} - 1$ ，测试数据不超过 100 组

思路1

设 $\mathrm{dp}[i][j]$ 表示恰有 i 个位以 j 开头的不降数字个数

初始时 $\mathrm{dp}[1][i] = 1$

当 $i > 1$ 时，仅需保证最高位最小

$$\mathrm{dp}[i][j] = \sum_{k=j}^{9} \mathrm{dp}[i-1][k]$$

考虑统计答案

#2869、数字游戏

从高往低考虑，令 pre 表示前一数位

- 填入 $\mathrm{pre} \sim \mathrm{num}_i - 1$

$$\mathrm{ans} \leftarrow \mathrm{ans} + \sum_{j=\mathrm{pre}}^{\mathrm{num}_i - 1} \mathrm{dp}[i][j]$$

- 填入 num_i

  若 $\mathrm{num}_i < \mathrm{pre}$ 无需继续讨论

  否则令 $\mathrm{pre} \leftarrow \mathrm{num}_i$ 继续讨论下一数位

当考虑到最低位时令 $\mathrm{ans} \leftarrow \mathrm{ans} + 1$

思路2

在搜索中增加一个参数 pre 表示前一数位

每一层从 pre ~ 上界 枚举对应数字

记忆化即可

#750、Windy数

题目描述

Windy 定义了一种 Windy 数:

> 不含前导零且相邻两个数字之差至少为 2 的正整数被称为 Windy 数

Windy 想知道在 A 和 B 之间,包括 A 和 B ,总共有多少个 Windy 数?

输入格式

一行两个数,分别为 A, B

输出格式

输出一个整数,表示答案

样例输入

```
1 10
```

样例输出

```
9
```

数据范围

对于 20% 的数据满足 $1 \le A \le B \le 10^6$

对于 100% 的数据满足 $1 \le A \le B \le 2 \times 10^9$

思路1

设 $dp[i][j]$ 表示恰有 i 个位以 j 开头的 windy 数

初始时 $dp[1][i] = 1$

当 $i > 1$ 时

$$dp[i][j] = \sum_{\substack{k=j \\ |j-k| \ge 2}}^{9} dp[i-1][k]$$

从高往低考虑

令 pre 表示前一数位,设有 cnt 个数位

不妨指定下界来避免前导 0,记 d 为当前数位的下界

若当前位为最高位时 $d = 1$ 否则 $d = 0$

若填入 $d \sim \ num_i - 1$ 即

#750、Windy数

$$\text{ans} \leftarrow \text{ans} + \sum_{\substack{j=d \\ |j-\text{pre}|\ge 2}}^{\text{num}_i - 1} \text{dp}[i][j]$$

若填入 num_i

若 $|j-\text{pre}|<2$ 无需继续讨论，否则 令 $\text{pre} \leftarrow \text{num}_i$ 继续讨论下一数位

当考虑到最低位时令 $\text{ans} \leftarrow \text{ans}+1$

上述枚举仅考虑了 cnt 位的情况，最后需累加不足 cnt 位的数量

$$\text{ans} \leftarrow \text{ans} + \sum_{i=1}^{\text{cnt}-1} \sum_{j=1}^{9} \text{dp}[i][j]$$

思路2

在搜索中增加参数 lead 表示是否存在前导零

枚举对应数字，忽略无前导 0 且相邻差值小于 2 的情况 ，记忆化即可

#754、数字计数

题目描述

给定两个正整数 a, b

求在 $[a, b]$ 中的所有整数中,每个数码 d 各出现了多少次

输入格式

仅包含一行两个整数 a, b ,含义如上所述

输出格式

包含一行 10 个整数,分别表示 $0 \sim 9$ 在 $[a, b]$ 中出现了多少次

样例输入

```
1 99
```

样例输出

```
9 20 20 20 20 20 20 20 20 20
```

数据范围

对于 30% 的数据中 $1 \le a \le b \le 10^6$

对于 100% 的数据中 $1 \le a \le b \le 10^{12}$

思路1

暂不考虑前导零，那么恰有 i 个数位时各数码出现的次数相同

设 $dp[i]$ 为恰有 i 位时每个数字出现的次数，有

$$dp[i] = 10 \times dp[i-1] + 10^{i-1}$$

- 10^{i-1} 为第 i 位的数字的贡献

  高位填入确定的某个数码，后续 $i-1$ 位有 10^{i-1} 个数

  每个数都可产生 1 的贡献

- $10 \times dp[i-1]$ 为后续 $i-1$ 位中数字的贡献

  如数字 1233 中的 3 在之前产生的 2 的贡献

  最高位分别填入 0~9，每个选择都产生 2 的贡献

  即 $dp[i-1]$ 中的每个都产生 10 的贡献

#754、数字计数

记 cnt_j 表示数码 j 出现次数，考虑求出 cnt_j

将数字从低到高逆序拆分存入 num 数组中

当前为从低到高第 i 位，考虑当前位上的数码

$$\overline{0\,0\ldots0\,0} \sim \overline{1\,0\ldots0\,0} - 1$$

$$\overline{1\,0\ldots0\,0} \sim \overline{2\,0\ldots0\,0} - 1$$

$$\cdots$$

$$\overline{(\text{num}_i - 1)\,0\ldots0\,0} \sim \overline{\text{num}_i\,0\ldots0\,0} - 1$$

当前位填入 $0 \sim \text{num}_i - 1$

对于当前位填入 j 时后续 10^{i-1} 个数都产生 1 的贡献

$$\text{cnt}_j \leftarrow \text{cnt}_j + 10^{i-1}$$

当前位填入 num_i

#754、数字计数

需考虑

$$\overline{\text{num}_i\ 0 \dots\ 0\ 0} \sim \overline{\text{num}_i\ \text{num}_{i-1} \dots\ \text{num}_2\ \text{num}_1}$$

即令

$$\text{cnt}_{\text{num}_i} \leftarrow \text{cnt}_{\text{num}_i} + \sum_{j=1}^{i-1}\left(\text{num}_i \times 10^{i-1-j}\right)$$

上述仅考虑了当前位数码的贡献，还需考虑 $1 \sim i-1$ 位上的数码贡献

对于每个数码，当前位确定后都有 $\text{dp}[i-1]$ 的贡献

即对于 $0 \le j \le 9$ 令

$$\text{cnt}_j \leftarrow \text{cnt}_j + \text{num}_i \times \text{dp}[i-1]$$

若高位至当前位都为 0 那么即为前导零，需虑减去贡献

#754、数字计数

后续位上共有 10^{i-1} 个数，即对于每一位都令

$$\mathrm{cnt}_0 \leftarrow \mathrm{cnt}_0 - 10^{i-1}$$

依次考虑每一位填入的数即可

思路2

对 $0 \sim 9$ 的数字逐一进行统计，设当前考虑 d 出现的次数

增加一个参数 lead 表示是否存在前导零，增加一个参数 t 表示 d 出现的次数

每层枚举当前位的数码

若有前导零且当前数码为 0

说明高位数码全为 0，不改变 t 枚举下一数位数位

否则累加数位 d 出现的次数

记忆化即可

#748、Amount of Degrees

题目描述

求给定区间 $[X, Y]$ 中满足下列条件的整数个数:

这个数恰好等于 K 个互不相等的 B 的整数次幂之和

如 $X = 15, Y = 20, K = 2, B = 2$,则有且仅有下列三个数满足题意:

$$17 = 2^4 + 2^0$$

$$18 = 2^4 + 2^1$$

$$20 = 2^4 + 2^2$$

输入格式

第一行包含两个整数 X 和 Y

接下来两行包含整数 K 和 B

输出格式

只包含一个整数

表示满足条件的数的个数

数据范围

对于全部数据 $1 \le X \le Y \le 2^{31} - 1, 1 \le K \le 20, 2 \le B \le 10$

要求不超过 X 的 B 进制下，数位 1 有 K 个的数量

将数以 B 进制进行拆分，从高位往低位考虑

记答案为 ans

思路1

与 #752 类似从高到低讨论每个数位的情况

即填入 $0 \sim \mathrm{num}_i - 1$ 和 num_i 两类情况

本题中合法的数字中仅能填入 0/1 两种数码

记已填入 1 的个数为 cnt

若 $\mathrm{num}_i = 0$

无需讨论 $0 \sim \mathrm{num}_i - 1$ 情况

#748、Amount of Degrees

若 $\mathrm{num}_i > 0$

- 填入 0

  剩余 $i-1$ 位选出 $\mathrm{K}-\mathrm{cnt}$ 位填入 1

  即令 $\mathrm{ans} \longleftarrow \mathrm{ans} + \binom{i-1}{\mathrm{K}-\mathrm{cnt}}$

  继续考虑下一数位

- 填入 1

  若 $\mathrm{num}_i = 1$

    此时令 $\mathrm{cnt} \longleftarrow \mathrm{cnt} + 1$

    若 $\mathrm{cnt} > \mathrm{K}$ 说明后续无需讨论，否则继续讨论下一数位

  若 $\mathrm{num}_i > 1$

    仅需从剩余 $i-1$ 位中选出 $\mathrm{K}-\mathrm{cnt}-1$ 个 1 即可

#748、Amount of Degrees

无需讨论下一数位

若继续考虑后续数位即当前位填入数码超过 1，都不合法

即令 $\text{ans} \longleftarrow \text{ans} + \binom{i-1}{\text{K}-\text{cnt}-1}$

当考虑到最低位且已填入 K 个 1 需令 $\text{ans} \longleftarrow \text{ans} + 1$

思路2

在搜索中增加一个参数 cnt ，表示已填入 1 的数量

若 $\text{num}_i > 1$

当前位可填入 0/1

若 $\text{num}_i = 0$

贴紧上界时，只能填入 0

未贴紧上界时，可填入 0/1

记忆化即可

二进制 & 位运算

在计算机中数以二进制的形式存储，根据该特点可用一个数表示一个集合

具体而言：通过第 i 个二进制位是否为 1 表示第 i 各元素是否包含在集合中

如 $(11)_{10} = (1011)_2$ 表示一个包含第 0、1、3 元素的集合

如下给出集合与集合的运算

集合	含义	集合示例	位运算	位运算示例
$A \cap B$	A，B 的交集	$\{0,2,3\} \cap \{0,1,2\} = \{0,2\}$	a & b	1101 & 0111 = 0101
$A \cup B$	A，B 的并集	$\{0,2,3\} \cup \{0,1,2\} = \{0,1,2,3\}$	a | b	1101 | 0111 = 1111
$A \Delta B$	A，B 的对称差	$\{0,2,3\} \Delta \{0,1,2\} = \{1,3\}$	$a \oplus b$	$1101 \oplus 0111 = 1010$
$A \setminus B$	A 与 B 的差	$\{0,2,3\} \setminus \{1,2\} = \{0,3\}$	a & ~b	1101 & 1001 = 1001
$A \setminus B, B \subseteq A$	B 为 A 的子集时 A 与 B 的差	$\{0,2,3\} \setminus \{0,2\} = \{3\}$	$a \oplus b$	$1101 \oplus 0101 = 1000$
$B \subseteq A$	B 包含于 A	$\{0,2\} \subseteq \{0,2,3\}$	a & b = b a | b = a	1101 & 0101 = 0101 1101 | 0101 = 1101

二进制 & 位运算

如下给出集合与元素的运算

集合	含义	集合示例	位运算	位运算示例
$\{x\}$	单个元素 x 构成的集合	$\{3\}$	$1 \ll x$	$1 \ll 3$
$U = \{0,1,\cdots,n-1\}$	全集	$\{0,1,2,3\}$	$(1 \ll n) - 1$	$(1 \ll 3) - 1$
$x \in S$	元素 x 属于集合 S	$2 \in \{0,2,3\}$	$S \,\&\, (1 \ll x) \neq 0$	$1101 \,\&\, 0100 = 0100$
$x \notin S$	元素 x 不属于集合 S	$1 \notin \{0,2,3\}$	$S \,\&\, (1 \ll x) = 0$	$1101 \,\&\, 0010 = 0000$
$S \cup \{x\}$	将元素 x 加入集合 S	$\{0,2,3\} \cup \{1\}$	$S \mid (1 \ll x)$	$1101 \mid 0010 = 1111$
$S \setminus \{x\}$	将元素 x 从集合 S 删除	$\{0,2,3\} \setminus \{2\}$	$S \,\&\, \sim(1 \ll x)$	$1101 \,\&\, 1011 = 1011$
$S \setminus \{x\}, x \in S$	将元素 x 从集合 S 删除(x 属于集合 S)	$\{0,2,3\} \setminus \{2\}$	$S \oplus (1 \ll x)$	$1101 \oplus 0100 = 1011$
	将最小元素从集合 S 删除		$S \,\&\, (S-1)$	$1101 \oplus 1100 = 1100$

二进制 & 内建函数

GCC 中还有一些用于位运算的内建函数:

- int __builtin_ffs(int x)

  返回 x 的二进制末尾最后一个 1 的位置，位置的编号从 1 开始 (最低位编号为 1)

  当 x 为 0 时返回 0

- int __builtin_clz(unsigned int x)

  返回 x 的二进制的前导 0 的个数

  当 x 为 0 时，结果未定义

- int __builtin_ctz(unsigned int x)

  返回 x 的二进制末尾连续 0 的个数

  当 x 为 0 时，结果未定义

- int __builtin_clrsb(int x)

二进制 & 内建函数

当 x 的符号位为 0 时返回 x 的二进制的前导 0 的个数减一

否则返回 x 的二进制的前导 1 的个数减一

- int __builtin_popcount(unsigned int x)

  返回 x 的二进制中 1 的个数

- int __builtin_parity(unsigned int x)

  判断 x 的二进制中 1 的个数的奇偶性

上述函数都可在末尾添加 l 或 ll (如 __builtin_popcountll) 来使参数类型变为 (unsigned) long 或 (unsigned) long long

返回值仍然是 int 类型

上述函数是内建函数经编译器高度优化，运行速度 **很快**

#3738、 数对

题目描述

给出一个长度为 N 的正整数序列 A 以及一个整数 K

对于序列中的两个数 x, y 若

- $x \in A \land y \in A$.
- x and y 与 x or y 两数二进制中 1 的数量之和不小于 K

那么称 x, y 称为好数对

请你统计 A 中有多少好数对

若 (a, b) 和数对 (c, d) 满足 $a \neq c$ 或 $b \neq d$ 则认为 (a, b) 和 (c, d) 不是相同的数对

输入格式

第一行输入两个空格分隔的整数 N, K

第二行输入 N 个空格分隔的整数表示序列 A

输出格式

输出一个整数表示答案

对于 x or y 和 x and y

若某位上都为 1 那么将产生两次贡献

若仅有一个 1 那么仅产生一次贡献

如 $x = 110, y = 011$

其中 010 将产生两次贡献

100 与 001 将分别产生一次贡献

数据规模

对于 8% 的数据 $1 \leq N \leq 20$

对于 20% 的数据 $1 \leq N \leq 5000$

对于 100% 的数据 $1 \leq N \leq 5 \times 10^5, 1 \leq A_i \leq 10^9, 1 \leq K \leq 60$

#3738、 数对

记 $\mathrm{b_x}$ 为 x 二进制中数位为 1 的数量，有

$$\mathrm{b}_{\mathrm{x\ \textbf{and}\ y}} + \mathrm{b}_{\mathrm{x\ \textbf{or}\ y}} = \mathrm{b_x} + \mathrm{b_y}$$

不妨将 $\mathrm{x, y}$ 视为集合 $\mathrm{A, B}$，根据容斥原理

$$|\mathrm{A} \cap \mathrm{B}| + |\mathrm{A} \cup \mathrm{B}| = |\mathrm{A}| + |\mathrm{B}|$$

从集合视角不难证明上述结论

问题转化为：统计 $\mathrm{b_x} + \mathrm{b_y} \geq \mathrm{k}$ 的数量

为避免重复统计将序列去重，再统计 $\mathrm{b_x}$ 的个数 cnt

若 $\mathrm{i + j \geq k}$ 根据乘法原理将 $\mathrm{cnt_i \times cnt_j}$ 计入贡献

显然 cnt 值域至多 30，不妨直接枚举 $\mathrm{i, j}$（也可前缀和优化）

若 A_i 值域为 V 时间复杂度 $O(n \log \mathrm{V})$

#2867、最短Hamilton路径

题目描述

给定一张 n 个点的带权无向图，点从 $0 \sim n-1$ 标号

求起点 0 到终点 $n-1$ 的最短 Hamilton 路径

Hamilton 路径的定义是从 $0 \sim n-1$ 不重不漏地经过每个点恰好一次

输入格式

第一行输入整数 n

接下来 n 行每行 n 个整数，其中第 i 行第 j 个整数表示点 i 到 j 的距离(记为 a_{ij})

输出格式

输出一个整数，表示最短 Hamilton 路径的长度

数据范围

对于全部的数据 $1 \le n \le 20, 0 \le a_{i,j} \le 10^7$

对于任意的 x, y, z ,数据保证 $a_{xx} = 0, a_{xy} = a_{yx}, a_{xy} + a_{yz} \ge a_{xz}$

最短路?全排列?

设 dp[u][S] 为从起点到达 u 点

且点经过状态为 S 的最短长度

其中 S 的第 i 个二进制位取 1 表示点 i 已经到达

$0 \sim 2^n - 1$ 枚举所有状态 S，考虑从点 v 走到 u

#2867、最短Hamilton路径

若状态 S 的第 u 个二进制位为 1 时考虑转移

记 $S_v = S \oplus (1 \ll v)$ 表示未经过 v 点的状态

若 S_v 中的第 v 个二进制位为 1 有

$$\mathrm{dp}[S][u] = \min(w[v][u] + \mathrm{dp}[v][S_v])$$

答案为 $\mathrm{dp}[n-1][2^n-1]$

时间复杂度 $O(n^2 \times 2^n)$

对于合法的状态 S 其必然最低位为 1 (合法状态必为从 0 出发) , 可排除一半的非法状态

适当考虑缓存命中率和数据的访问顺序可能显著影响程序的性能

```
memset(dp, 0x3f, sizeof dp);
dp[1][0] = 0;
for (int s = 1; s < 1 << n; s += 2)
  for (int i = 0; i < n; i++)
    for (int j = 0; (s >> i & 1) && j < n; j++)
      if (s >> j & 1)
        dp[s][i] = min(dp[s][i], dp[s ^ 1 << i][j] + w[j][i]);
```

#3979、 回文删除

题目描述

给出一个长度为 n 仅由小写字母组成的字符串 S

允许进行如下操作

- 删除 S 的一个回文**子序列**
- 将删除后的结果作为新的 S

请你计算最少需要几次操作才能将 S 删为空串

输入格式

第一行输入一个整数 T ，表示 T 组询问

每组询问输入一行一个字符串 S

输出格式

每组询问输出一个整数

数据规模

对于 25% 的数据 $1 \leq |S| \leq 12$

对于全部的数据 $1 \leq |S| \leq 16, 1 \leq T \leq 5$

记串长度为 n

将序列以二进制形式描述

如串为 “**abb**” 那么 $S = (101)$ 表示子序列 “**ab**”

令 dp[S] 表示序列状态为 S 时的最少删除次数

答案为 $dp[2^n - 1]$

#3979、 回文删除

初始时 $dp[0]=0$ ，若 S 对应子序列为回文串时 $dp[S]=1$ 否则 $dp[S]=\infty$

对于 S 找出其子集 S’

$$dp[S]=\min_{S'\subset S}(dp[S']+dp[S\oplus S])$$

若直接枚举 S, S’ 验证 S’ ⊂ S 时间复杂度 $O(4^n)$

给出一种更高效的实现方式

若 S 在二进制下结构为 $(111\cdots1111)_2$ 每次令 $S\leftarrow S-1$ 即可枚举其所有子集

每次都可令最右侧的 1 变为 0 , 其更低位都变为 1 容易证明其正确性

若 S 在最高位后存在 0 令 $S'=S$ 每次令 $S'\leftarrow S'-1$ 再令 T & N 即可

仅令 $T\leftarrow T-1$ 可能使得 S 中为 0 的位变为 1

如 $S'=S=(1010)_2$

$$S'=\overbrace{1010-0001}^{1001}\ \&\ 1010=1000$$

#3979、 回文删除

$$S' = \overbrace{1010\ - 0001}^{0111} \,\&\, 1010 = 0010$$

$$S' = \overbrace{0010\ - 0001}^{0000} \,\&\, 1010 = 0001$$

若将 1 紧凑排布，即为二进制减法

紧凑排布的减法于上述过程构成一一映射，可保证其正确性

对于枚举 S 的子集 S’ 时间复杂度为 $O(2^{|S|})$

$|S| \in [0, n]$ 且元素个数为 $|S|$ 的集合共有 $\binom{n}{|S|}$ 个，根据 二项式定理

$$\left(\sum_{S \subset \{0,1,2,\cdots,n-1\}} 2^{|S|}\right) = \sum_{i=0}^{n}\left(\binom{n}{i}2^i\right) = 3^n$$

总时间复杂度 $O(3^n)$

提前预处理 S 的子集并仅在 S’ 与 S 间转移 时间复杂度也为 $O(3^n)$，但常数较大

#2868、蒙德里安的梦想

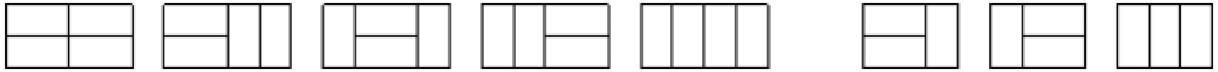
题目描述

求把 $N \times M$ 的棋盘分割成若干个 1×2 的长方形，有多少种方案

例如当 $N = 2, M = 4$ 时,共有 5 种方案

当 $N = 2, M = 3$ 时,共有 3 种方案

如下图所示:



输入格式

输入包含多组测试用例

每组测试用例占一行,包含两个整数 N, M

当输入用例 $N = M = 0$ 时,表示输入终止,且该用例无需处理

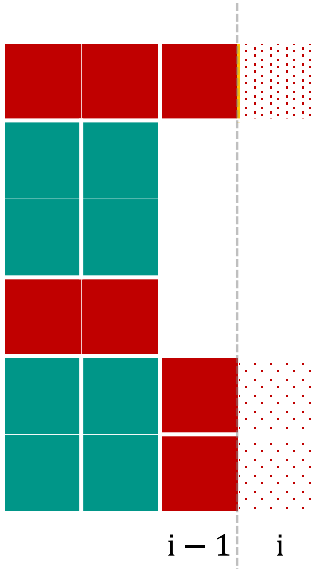
输出格式

每个测试用例输出一个结果

每个结果占一行

数据范围

对于全部的数据 $1 \le N, M \le 11$



考虑以列为分隔，可将整个棋盘分为两个部分

右图中红色部分为从 $i-1$ 列伸出到第 i 列的部分

这些伸出的部分决定了下一列必须补全该砖块

剩下的空白部分，对后序分隔无影响

仅考虑横着放入的方式，剩余部分放入方式 **唯一确定**

#2868、蒙德里安的梦想

考虑以列作为阶段

设 $\mathrm{dp}[i][\mathrm{S}]$ 为考虑到第 i 列放置状态为 S 时的方案数

其中 S 的第 i 个二进制位表示第 i 行伸出

第 i 列的 S 状态能够从 $i-1$ 行 S' 状态转移，仅当

- $\mathrm{S}\ \&\ \mathrm{S}' = 0$

  保证伸出部分不能再被放置

- $\mathrm{S} \mid \mathrm{K}$ 中不存在奇数个连续的 0

  若部分空出区域由砖块横着填充，该状态与本阶段其它状态重合

  空出区域只由砖块竖着填充，连续奇数个无法做到

答案为 $\mathrm{dp}[m][0]$

时间复杂度 $O(2^n \times 2^n \times m) = O(4^n \times \mathrm{m})$

#755、互不侵犯

题目描述

在 $n \times n$ 的棋盘上放 k 个国王

国王可攻击相邻的 8 个格子

求使它们无法互相攻击的方案总数

输入格式

只有一行,包含两个整数 n, k

输出格式

输出方案总数,若不能够放置则输出 0

样例输入1

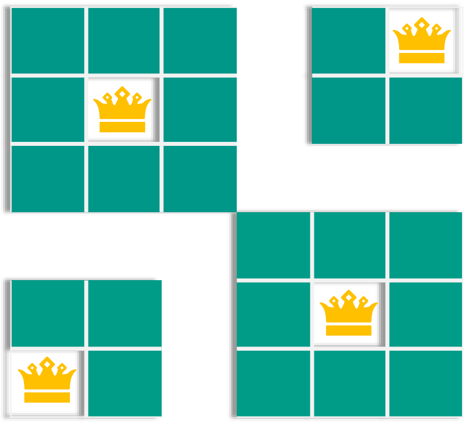
```
3 2
```

样例输出1

```
16
```

数据范围

对于全部数据 $1 \le n \le 10, 0 \le k \le n^2$



从上往下逐行考虑放置国王

若第 i 列放置国王，其仅受到第 $i-1$ 行限制

考虑将行作为阶段，进行转移

设 $dp[S][i][j]$ 表示考虑到第 i 行共放置了j个国王

第 i 行国王放置状态为 S 的方案数

其中 S 的第 i 个二进制位为 1 表示第 i 列摆放国王

#755、互不侵犯

对于合法的状态 S 应当满足 S 中不存在相邻的二进制 1

对于两个摆放状态 S_1, S_2 能够转移，仅当

- $S_1 \ \& \ S_2 = 0$

  即相邻两行的相同列不能同时为 1，不能同时摆放国王

- $S_1 \mid S_2$ 中不存在相邻的二进制1

记 cnt_s 为状态 S 中的二进制 1 的数量，所有合法能转移到 S 的状态集合为 V_S

$$\mathrm{dp}[S][i][j] = \sum_{v \in V_S} \mathrm{dp}[v][i-1][j - \mathrm{cnt}_s]$$

若直接枚举 S 及 v 再验证能否转移，时间复杂度 $O(k \times n^3 \times 4^n)$

预先处理出所有合法状态和集合 V_s 即可，答案为

$$\left(\sum_{s} \mathrm{dp}[S][n][k]\right) = \mathrm{dp}[0][n+1][k]$$

实验舱
青少年编程
走近科学 走进名校

谢谢观看