

谜题 II (puzzle)

【题目描述】

厌倦了计算一些稀奇古怪的问题的答案对 998244353 或 $10^9 + 7$ 取模后的结果，小 E 设计了一道与 2 相关的谜题：

给定一个 2 行 2^k 列的网格，你需要将 $1 \sim 2^{k+1}$ 不重不漏地填入网格的 2^{k+1} 个格子中，并使得相邻格子之间的大小关系满足下列限制。

具体地，给定三个长度分别为 $2^k - 1, 2^k, 2^k - 1$ 的 01 序列 a, b, c 。称一个 2×2^k 的矩阵 A 为该谜题的一组解，当且仅当其同时满足以下四条约束：

1. A 包含 $1, 2, \dots, 2^{k+1}$ 中的每个整数恰好一次；
2. 对于所有 $1 \leq i < 2^k$ ，均有 $a_i = [A_{1,i} < A_{1,i+1}]$ ；
3. 对于所有 $1 \leq i \leq 2^k$ ，均有 $b_i = [A_{1,i} < A_{2,i}]$ ；
4. 对于所有 $1 \leq i < 2^k$ ，均有 $c_i = [A_{2,i} < A_{2,i+1}]$ 。

其中 $[P]$ 当条件 P 成立时取值为 1，否则为 0。

对你而言，仅仅找出任意一组解或判断无解太过简单。既然是关于 2 的谜题，小 E 要求你计算谜题的解的数量对 2 取模后的结果。

小 E 准备了 t 道网格大小相同的谜题，并使用高效的压缩手段将它们压缩在一起。你需要对于其中每一道谜题，求出解的数量对 2 取模后的结果。

【实现细节】

选手不需要，也不应该实现 main 函数。

选手需要确保提交的程序包含头文件 puzzle.h，即在程序开头加入以下代码：

```
1 #include "puzzle.h"
```

选手需要在提交的程序源文件中实现以下函数：

```
1 unsigned puzzle(int t, int k, std::vector<unsigned> a,
  std::vector<unsigned> b, std::vector<unsigned> c);
```

- t, k 分别表示谜题的个数与网格的大小。
- 对于 $0 \leq i < 2^k - 1$ ， a_i 二进制下的第 j ($0 \leq j < t$) 位表示第 $j + 1$ 个谜题中 $A_{1,i+1}$ 与 $A_{1,i+2}$ 的大小关系的限制。
- 对于 $0 \leq i < 2^k$ ， b_i 二进制下的第 j ($0 \leq j < t$) 位表示第 $j + 1$ 个谜题中 $A_{1,i+1}$ 与 $A_{2,i+1}$ 的大小关系的限制。
- 对于 $0 \leq i < 2^k - 1$ ， c_i 二进制下的第 j ($0 \leq j < t$) 位表示第 $j + 1$ 个谜题中 $A_{2,i+1}$ 与 $A_{2,i+2}$ 的大小关系的限制。
- 该函数需要返回一个非负整数，其中二进制下的第 j ($0 \leq j < t$) 位表示第 $j + 1$ 个谜题中解的数量对 2 取模后的结果。

- 对于每个测试点，该函数会被交互库调用恰好一次。

注意：在任何情况下，交互库运行所需时间均不会超过 0.1 秒，所用内存为固定大小，且均不超过 64 MiB。

【测试程序方式】

试题目录下的 `grader.cpp` 是交互库参考实现，最终测试时所用的交互库实现与该参考实现有所不同，因此选手的解法不应该依赖交互库实现。

选手可以在本题目录下使用如下命令编译得到可执行程序：

```
1 g++ grader.cpp puzzle.cpp -o puzzle -O2 -std=c++14 -static
```

对于编译得到的可执行程序：

- 可执行文件将从标准输入读入以下格式的数据：
 - 输入的第一行包含两个正整数 t, k ，分别表示谜题的数量与网格的大小。
 - 输入的第 $3i-1$ ($1 \leq i \leq t$) 行包含一个长度为 2^k-1 的 01 字符串 $a_1 \dots a_{2^k-1}$ 。
 - 输入的第 $3i$ ($1 \leq i \leq t$) 行包含一个长度为 2^k 的 01 字符串 $b_1 \dots b_{2^k}$ 。
 - 输入的第 $3i+1$ ($1 \leq i \leq t$) 行包含一个长度为 2^k-1 的 01 字符串 $c_1 \dots c_{2^k-1}$ 。
- 可执行文件将输出以下格式的数据至标准输出：
 - 输出共 t 行，其中第 i ($1 \leq i \leq t$) 行包含一个非负整数，表示第 i 道谜题的解的数量对 2 取模后的结果。

【样例 1 输入】

```
1 2 1
2 1
3 01
4 1
5 1
6 00
7 1
```

【样例 1 输出】

```
1 1
2 0
```

【样例 2 输入】

```
1 4 2
2 100
3 1111
4 100
5 100
6 0000
7 110
8 110
9 1111
10 011
11 010
12 0000
13 010
```

【样例 2 输出】

```
1 1
2 1
3 0
4 0
```

【附加文件说明】

在附加文件中：

1. `grader.cpp` 是提供的交互库参考实现。
2. `puzzle.h` 是头文件，选手不用关心具体内容。
3. `template_puzzle.cpp` 是提供的示例代码，选手可参考并实现自己的代码。

【子任务】

对于所有测试数据，均有：

- $1 \leq t \leq 32$, $1 \leq k \leq 18$;
- 对于所有 $1 \leq i < 2^k$, 均有 $a_i \in \{0, 1\}$;
- 对于所有 $1 \leq i \leq 2^k$, 均有 $b_i \in \{0, 1\}$;
- 对于所有 $1 \leq i < 2^k$, 均有 $c_i \in \{0, 1\}$ 。

子任务编号	分值	$k \leq$	特殊性质
1	5	2	无
2		3	
3		6	
4		8	
5		10	
6	25	13	A
7	5	14	无
8	20	17	A
9	10	18	B
10	15		无

特殊性质 A: $t \leq 4$ 。

特殊性质 B: 对于所有 $1 \leq i < 2^k$, 均有 $b_i \neq b_{i+1}$ 。

【评分方式】

注意:

- 选手不应当通过非法方式获取交互库的内部信息, 如直接与标准输入、输出流进行交互。此类行为将被视为作弊;
- 最终的评测交互库与样例交互库的实现不同。

本题首先会受到和传统题相同的限制, 例如编译错误会导致整道题目得 0 分, 运行时错误、超过时间限制、超过空间限制等会导致相应测试点得 0 分等。选手只能在程序中访问自己定义的变量以及交互库给出的变量, 尝试访问其他地址空间将可能导致编译错误或运行错误。

在上述条件基础上:

- 对于每个测试点, 程序得到满分当且仅当调用 `puzzle` 函数时返回的答案正确。