提高组400+试题 第五组-新

中文题目名称	选数字	堆箱子	快速排序	统计学带师
英文题目名称	select	place	sort	statistics
每个测试点建议时限	2000	1000	1000	1000
每个测试点空间限制	256 M	256 M	128 M	128 M
测试点数目	20	25	20	20
每个测试点分值	5	4	5	5
比较方式	逐行比较	逐行比较	逐行比较	逐行比较
浮点输出误差精度	-	-	-	-

注意：

- 英文题目名称即文件名，若文件名为 filename，则提交的文件为filename.pas/c/cpp，程序输入输出文件名分别为 filename.in filename.out。
- 建议时限仅供参考，具体按照评测机上标程运行时间的2 - 3倍设置。
- 建议将栈大小设为64m，并打开编译参数O2。

选数字

题目限制

2000 ms 256 M

题目描述

桌面上有N个数字排成一排，小武要求小林从中选出3个数字，使得这3个数字的按位或的结果恰好等于x，小林很快就解决了这个问题，小武想了想，决定把问题加强一下，小武会问小林Q次问题，每次选的3个数字只能在从左往右的第l个数和第r个数之间选择，并且要小林说出符合要求的方案数，小林顿时不会了，于是把问题交给了你。

输入格式

第一行两个数N和Q
第二行N个数，按照从左到右的顺序给出桌面上的数字
接下来Q行，每行3个数字，分别为l,r,x

输出格式

Q行，每行一个数表示方案数

数据范围

对于20%的数据，n,Q≤100
对于60%的数据，n,Q≤10000
对于100%的数据，1≤l≤r≤n≤10^5,1≤Q≤10^5,1≤桌面上每个数字,x≤255

输入样例

```
10 5
2 4 3 7 6 9 8 7 10 15
1 5 7
2 8 14
3 5 7
1 10 15
6 9 12
```

输出样例

```
9
1
1
81
0
```

样例解释

第一个询问，选择范围为{2,4,3,7,6}。除了选2 4 6不行，剩下9种方案均可以

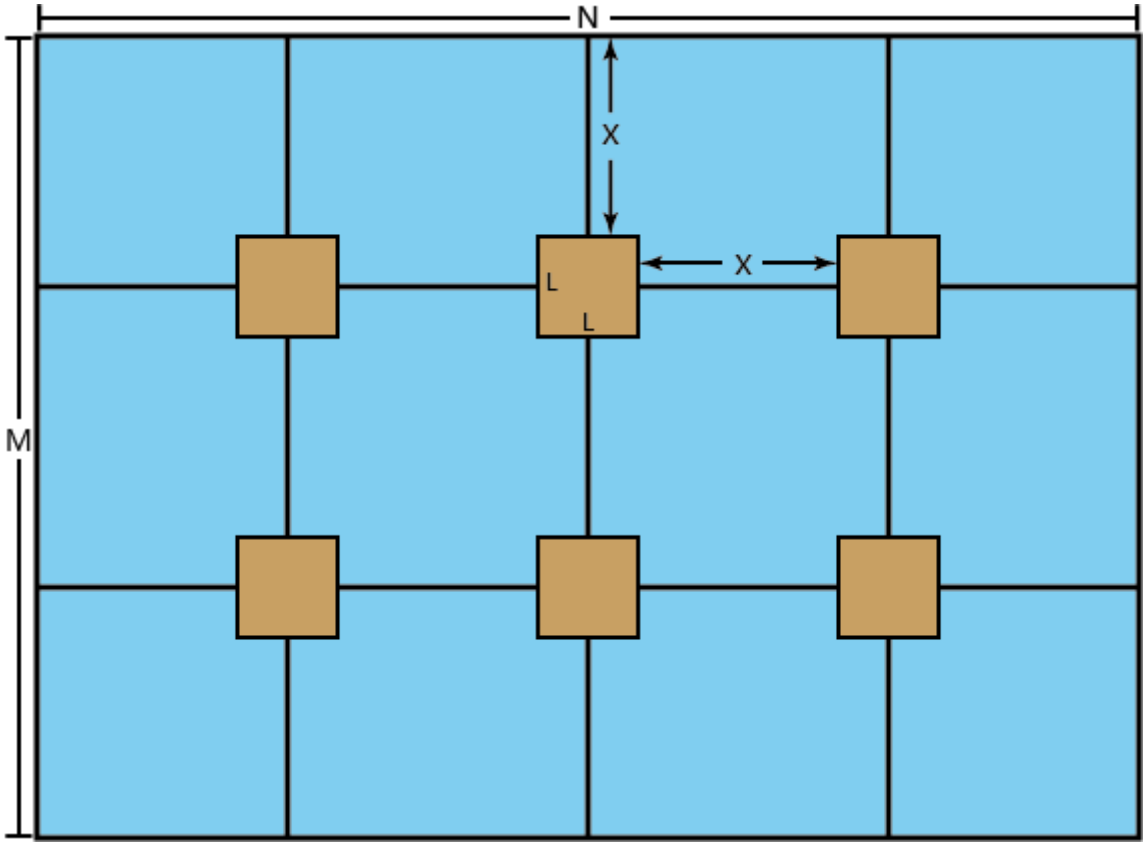
堆箱子

题目限制

1000 ms 256 M

题目描述

仓库中有很多的箱子，这天小武想把箱子收拾一下。仓库是一个长方体，我们不用考虑仓库的高度，仓库的长度为N，宽度为M，箱子是一个标准的正方体，边长为L。小武很严格，对于箱子的摆放有很严格的要求。小武要求用若干根直线等分仓库的长和宽，而箱子只能放在等分长和宽的直线的交点上，并且一个箱子到四个方向的距离（可能是到另一个箱子， 也可能是到仓库的边缘），必须相同。那么在满足要求的前提下，小武想放心尽可能多的箱子，那么此时一个箱子到四个方向的距离是多少呢？（设这个距离为x）



输入格式

一行三个数表示L，N，M

输出格式

一行一个实数表示距离x，精确到小数点后五位
如果无法满足要求，输出-1

数据范围

对于24%的数据，1≤L≤100，1≤N,M≤1000
对于48%的数据，1≤L≤1000，1≤L,N,M≤10^6
对于80%的数据，1≤L≤10^6
对于100%的数据，1≤L,N,M≤10^9

输入样例

```
样例1
2 18 13
样例2
4 26 26
样例3
3 46 18
```

输出样例

```
样例1
0.50000
样例2
0.28571
样例3
0.50000
```

样例解释

对于样例1，横排放7个箱子，竖排放5个箱子，x=(18-2*7)/8=(13-2*5)/6=0.50000

快速排序

题目限制

1000 ms 128 M

题目描述

小武最近学习了快速排序，写出了下述的代码

```
1   void Qsort(int a[], int low, int high)
2       {
3           if(low >= high) return;
4           int first = low;
5           int last = high;
6           int key_index = (rand() % (high - low + 1)) + low;
7           swap(a[first], a[key_index]);
8           int key = a[first];     /*用数组的第一个记录作为中枢*/
9           while(first < last) {
10              while(first < last && a[last] >= key) --last;
11              a[first] = a[last];             /*将比第一个小的移到低端*/
12              while(first < last && a[first] <= key) ++first;
13              a[last] = a[first];             /*将比第一个大的移到高端*/
14          }
15          a[first] = key;          /*中枢记录到位*/
16          Qsort(a, low, first - 1);
17          Qsort(a, first + 1, high);
18      }
19
```

这里开始low=0，high=N，数组a为1-N的一个排列

我们采用随机优化的快速排序是很难碰到最坏情况的，但是小林偷偷修改了运行环境，控制了随机数的生成，使得随机数依次为a1,a2,a3,...,ak,a1,...,即随机数结果依次为a1到ak，然后不断循环。但是还有一个问题，什么样的排列在这样的随机数下效果最差呢，小林认为效果最差即递归的深度最深，但小林不知道怎么找到这个排列，只好交给了你

输入格式

第一行两个整数N和k
接下来k行，每行一个数字，表示a1到ak

输出格式

N行整数，为1-N的一个排列，若有多个排列满足条件，输出其中字典序最小的那个排列

数据范围

对于40%的数据，1=N,k<=10
对于70%的数据，1<=N,k<=10000
对于100%的数据，1<=N,k<=50000, 0<=ai<=10^9

输入样例

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

输出样例

```
样例1
1
2
3
样例2
1
4
2
3
样例3
1
```

样例解释

对于样例1

(1,2,3) 递归深度为3层，字典序最小

对于样例2

(1, 4, 2, 3)递归深度为4层，字典序最小

(1,2,3,4) -> (1) 2 (3,4) -> 1 2 3 (4) 字典序小，但是只有3层深度

统计学带师

题目限制

1000 ms 128 M

题目描述

小 H 立志成为统计学带师。

这天他捡来了 n 个字符串，并把它们按照他喜欢的顺序排放好，得到一个字符串序列 $s_{1\ldots n}$ 。

小 L 为了帮助他实现梦想，决定对他策划一场训练。

小 L 也掏出了一个字符串 S ，并决定问小 H q 个问题。

第 i 个问题可以用三个正整数 l, r, k 描述，表示 $S_{l\ldots r}$ 在 $s_{1\ldots n}$ 每个字符串中出现次数的第 k 小值。

其中 $S_{l\ldots r}$ 表示 S 的所有下标在 $[l, r]$ 内的字符按照原顺序组成的字符串，如 $S = \texttt{abcbc}, S_{2\ldots 4} = \texttt{bcb}$ 。

由于小 L 自己并不是统计学带师，所以他在训练小 H 之前，希望你帮他求出标准答案。

输入格式

第一行，两个正整数 n, q。
第二行，一个字符串 S。
以下 n 行，第 i 行一个字符串 s_i。
以下 q 行，每行三个正整数 l, r, k。

输出格式

q 行，每行一个整数，表示小 L 的第 i 个问题的答案。

数据范围

对于 40% 的数据，n, q, |S|, ∑|s_i| <= 10^3；
对于 100% 的数据，1 <= n, |S|, ∑|s_i| <= 10^5, 1 <= q <= 2 × 10^5, 1 <= l <= r <= |S|, 1 <= k <= n，所有字符串均仅包含小写字母。

输入样例

```
5 4
abcbcq
abc
bcd
acbc
qcbcba
qwqwq
1 2 2
3 5 4
2 3 3
6 6 1
```

输出样例

```
0
1
1
0
```

样例解释

对于第一个问题，$S_{1\ldots2}=$ `ab` ，在 $s_{1\ldots n}$ 中的出现次数分别为 $1,0,0,0,0$ ，故第 2 小值为 0 。

对于第二个问题，$S_{3\ldots5}=$ `cbc` ，在 $s_{1\ldots n}$ 中的出现次数分别为 $0,0,1,1,0$ ，故第 4 小值为 1 。

对于第三个问题，$S_{2\ldots3}=$ `bc` ，在 $s_{1\ldots n}$ 中的出现次数分别为 $1,1,1,1,0$ ，故第 3 小值为 1 。

对于第四个问题，$S_{6\ldots6}=$ `q` ，在 $s_{1\ldots n}$ 中的出现次数分别为 $0,0,0,0,1$ ，故第 1 小值为 0 。