

# 半期测试

测试时间 2022年5月4日 8: 00 ~ 12: 00

题目	数三角形	装备强化	进位统计	最小生成树	最短路
文件名	count.cpp	strengthen.cpp	carry.cpp	tree.cpp	path.cpp
输入输出	count.in/.out	strengthen.in/.out	carry.in/.out	tree.in/.out	path.in/.out
时间限制	1s	1s	1s	1s	2s
空间限制	256MB	256MB	256MB	256MB	512MB
编译命令	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14	-O2 -std=c++14

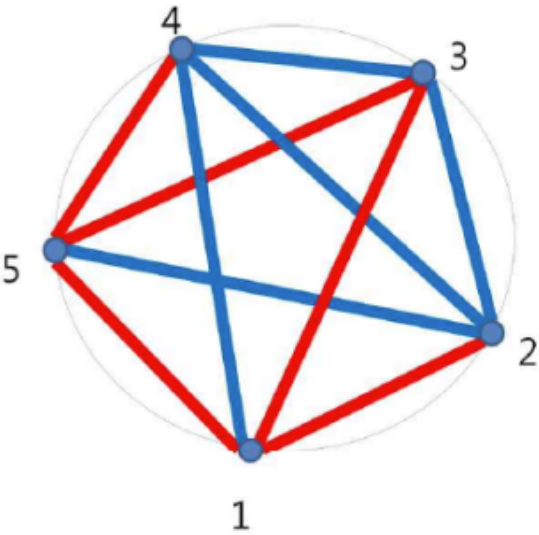
# T1: 数三角形(count)

文件名 `count.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

## 问题描述

$n$  个点平均分布在圆周上, 任意两点之间均有连边, 边只有红蓝两种颜色, 用 `1` 和 `0` 来表示。求出图中有多少个**纯色**三角形。

下图描述了样例1的情形, 图中只有 2 个纯色三角形: 蓝色  $\triangle 234$ , 红色  $\triangle 135$ 。



## 输入格式

第一行输入 1 个正整数  $n$ 。

接下来  $n - 1$  行, 第  $i$  行输入长度为  $n - i$  的 `01` 字符串, 表示点  $i$  与点  $i + 1 \sim n$  之间边的颜色, 用 `1` 表示红色, `0` 表示蓝色。

## 输出格式

输出一个整数, 表示图中纯色三角形的个数。

## 样例1

样例输入

```
5
1101
000
01
1
```

样例输出

## 样例2

见下发文件 `count2.in/.out`

## 数据范围

对于 30% 的数据,  $1 \leq n \leq 500$

对于 60% 的数据,  $1 \leq n \leq 2000$

另有 20% 的数据,  $a_{ij} = 1$

对于 100% 的数据,  $1 \leq n \leq 5000$

**温馨提示：**本题读入数据量较大，请一次性读取一行，或使用其它效率较高的读入手段。

# T2: 装备强化(strengthen)

文件名 `strengthen.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

## 问题描述

你在某款游戏中强化某件装备, 该装备的强化等级最高为  $n$  , 初始为  $0$  。

每次强化, 有  $p$  的概率强化成功, 使装备强化等级上升  $1$ ; 强化失败时, 装备强化等级会下降  $1$  。

当装备强化等级为  $0$  时, 强化失败不会继续降级。当装备强化等级达到最高级  $n$  时无法再强化。

请问将这件装备从  $0$  强化到  $n$  需要的期望强化次数为多少?

## 输入格式

输入仅包含  $1$  个正整数  $n$  和  $1$  个浮点数  $p$  。

## 输出格式

输出  $1$  个浮点数, 表示期望强化次数。

绝对误差或相对误差不超过  $10^{-6}$  。

## 样例1

样例输入

5 0.6

样例输出

16.316872427984

## 样例2

样例输入

3 0.2

样例输出

135

## 样例3

样例输入

123456 0.9

样例输出

154319.84375

数据范围

对于 30% 的数据,  $2 \leq n \leq 100$

对于 60% 的数据,  $2 \leq n \leq 10^5$

对于 100% 的数据,  $2 \leq n \leq 10^{18}$ ,  $0 < p < 1$ ,  $p$  小数点后最多包含 3 位。

数据保证答案的数值不超过  $10^{20}$ 。

# T3: 进位统计(carry)

文件名 `carry.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

## 问题描述

两个 10 进制非负整数的加法, 可以列竖式计算, 如下图所示:

$$\begin{array}{r} 658 \\ + 743 \\ \hline 1401 \end{array}$$

在这个例子中, 发生了 3 次进位。

现在有  $n$  个数  $a_1, a_2, \dots, a_n$  , 记  $f(a_i, a_j)$  为  $a_i + a_j$  的进位次数, 求:

$$\sum_{i=1}^n \sum_{j=i+1}^n f(a_i, a_j)$$

## 输入格式

第一行输入 1 个整数  $n$  。

第二行输入  $n$  个整数  $a_1, a_2, \dots, a_n$  。

## 输出格式

输出一个整数, 表示题目要求的答案。

## 样例1

样例输入

```
10
0 1 2 3 4 5 6 7 8 9
```

样例输出

```
20
```

## 样例2

见下发文件 `carry2.in/.out`

### 样例3

见下发文件 `carry3.in/.out`

### 数据范围

对于 30% 的数据,  $1 \leq n \leq 2000$

另有 20% 的数据,  $0 \leq a_i \leq 5000$

对于 100% 的数据,  $1 \leq n \leq 10^5$ ,  $0 \leq a_i \leq 10^9$

## T4: 最小生成树(tree)

文件名 `tree.cpp/.in/.out` , 时间限制1秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

### 问题描述

$n$  个点的完全图中有  $n^{n-2}$  种生成树。

生成树的权值定义为  $\sum_{i=1}^n f_{d_i}$  , 其中  $d_i$  表示点  $i$  在生成树中的度数。

给定  $n$  和  $f_1, f_2, \dots, f_{n-1}$  , 求权值最小的生成树的权值。

### 输入格式

第一行输入 1 个整数  $n$  。

第二行输入  $n - 1$  个整数  $f_1, f_2, \dots, f_{n-1}$  。

### 输出格式

输出一个整数, 表示最小的权值。

### 样例1

样例输入

```
4
5 1 4
```

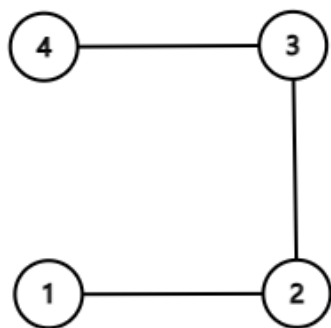
样例输出

```
12
```

样例解释

下图的方案中, 点 1、4 的度数为 1 , 点 2、3 的度数为 2, 权值之和为  $2 \times 5 + 2 \times 1 = 12$





## 样例2

见下发文件 `tree2.in/.out`

## 样例3

见下发文件 `tree3.in/.out`

## 数据范围

对于 30% 的数据,  $1 \leq n \leq 8$

另有 20% 的数据,  $f_i$  在  $[0, n]$  内等概率随机生成。

对于 100% 的数据,  $1 \leq n \leq 5000$ ,  $0 \leq f_i \leq 10^5$

# T5: 最短路(path)

文件名 `path.cpp/.in/.out` , 时间限制2秒, 空间限制512MB, 编译开启 `-O2` 优化, 支持 C++14。

## 问题描述

给出  $n$  个点  $m$  条双向边的无向图。

现在有  $q$  次 **独立的** 修改操作, 每次操作会修改 1 条边的长度, 在每次修改后, 输出 1 到  $n$  的最短路。

## 输入格式

第一行输入 3 个整数  $n, m, q$ 。

接下来  $m$  行, 每行输入 3 个整数  $u_i, v_i, w_i$  , 描述一条连接  $u_i$  与  $v_i$  长度为  $w_i$  的双向边。

再接下来  $q$  行, 每行输入 2 个整数  $x_i, y_i$  , 表示修改第  $x_i$  条边的长度为  $y_i$ 。边按输入的顺序从 1 开始编号。

数据保证从 1 出发能够到达  $n$ 。

## 输出格式

每次修改操作后, 输出 1 到  $n$  的最短路径长度。

注意, 每次修改操作是独立的, 也就是说, 在询问完毕后, 这次修改会撤销, 图的状态会恢复到原始状态。

## 样例1

样例输入

```
2 4 4
1 2 2
1 2 3
1 2 4
1 2 5
2 1
3 2
4 3
1 5
```

样例输出

```
1
2
2
3
```

## 样例2

样例输入

```
4 5 6
1 2 2
2 4 3
1 4 7
1 3 1
3 4 5
3 4
5 1
3 8
1 4
2 1
3 1
```

样例输出

```
4
2
5
6
3
1
```

样例3

见下发文件 path3.in/.out

样例4

见下发文件 path4.in/.out

数据范围

对于 30% 的数据,  $1 \leq n, m, q \leq 2000$

另有 30% 的数据,  $y_i \leq w_{x_i}$ , 即修改操作**不会**把边长增大

对于 100% 的数据,  $2 \leq n \leq 2 \times 10^5, 1 \leq m, q \leq 2 \times 10^5, 1 \leq w_i, y_i \leq 10^9$   
 $1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq x_i \leq m$