

模拟赛 3

按照 NOIP 比赛要求建立文件夹, 1/31 晚 8 点前发到邮箱 wushenguang@sina.com

共 3 道题目, 时间 4 小时

题目名称	robot	money	chance
文件名	robot	money	chance
输入文件	robot.in	money.in	chance.in
输出文件	robot.out	money.out	chance.out
时间限制	15s	10s	40s
内存限制	128mb	64mb	256mb
测试点数目	20	20	20
测试点分值	5	5	5
是否有部分分	否	否	否
题目类型	传统	传统	传统

评测环境

操作系统: Windows XP Professional SP3

CPU: Intel(R) Pentium(R) CPU G630 @ 2.70GHz (2CPUs)

系统内存: 2GB

评测工具: Cena 0.8.1

Problem 1 robot(robot.cpp/c/pas)

【题目描述】

VRI (Voltron 机器人学会) 的工程师建造了 n 个机器人。任意两个兼容的机器人站在同一个格子时可以合并为一个复合机器人。我们把机器人用 1 至 n 编号 ($n \leq 9$)。如果两个机器人的编号是连续的, 那么它们是兼容的, 可以合并成一个复合机器人。最初这 n 个机器人各自都只有唯一的编号。而一个由两个或以上的机器人合并构成的复合机器人拥有两个编号, 分别是构成它的所有机器人中最小和最大的编号。例如, 2 号机器人只可以与 1 号或 3 号机器人合并。若 2 号机器人与 3 号机器人合并, 可构成编号为 2-3 的复合机器人。如果编号为 2-3 的复合机器人与编号为 4-6 的复合机器人合并, 可构成编号为 2-6 的复合机器人。当所有机器人合并以后则构成 1- n 复合机器人。工程师把这 n 个机器人放在了一个封闭的房间中, 房间四周均是墙。该房间被划分成 $w \times h$ 个方格。有些方格有障碍物, 机器人不可经过或停留; 其余方格允许多个机器人停留, 同时允许机器人经过。任何时候一个机器人只占用一个方格。初始时刻, 所有机器人均在不同的方格中。这些原始的机器人不会自发地移动。它们只有被工程师沿 x 轴或 y 轴推动后, 才会沿推动的方向不断向前直线移动, 直至碰到障碍物或墙停止移动。停止移动后, 它会扫描当前的格子是否存在可以与它合并的机器人, 如果有, 则合并并继续检查, 直至不能再合并为止。工程师只能沿水平向左、水平向右、竖直向上、竖直向下四个方向推动机器人, 并且, 在机器人尚未停止移动时, 不允许推动其它机器人, 因此任何时刻, 房间中都只能有一个机器人移动, 为了帮助机器人转向, 工程师在一些格子

中放置了转向器。具体地说，转向器分为顺时针转向器（右转器）和逆时针转向器（左转器），顺时针转向器可以使到达该格子的机器人沿顺时针方向转向 90°；逆时针转向器可以使到达该格子的机器人沿逆时针方向转向 90°。现在，我们将告诉你初始时刻房间内的信息。请你计算工程师最少共计需要推动机器人多少次，才能把所有的 n 个机器人全部合并（如果可能的话）。

【输入格式】

你的程序必须从标准输入读入。输入的第 1 行包含 3 个整数 n 、 w 和 h ，用空格隔开。输入文件中接下来的 h 行描述初始时刻房间内的信息，每行包含 w 个字符。这 $w \times h$ 字符中每一个表示房间中的一个格子，意义如下：

‘1’至‘9’：表示该方格中有一个机器人，编号为这个数字；

‘x’：表示该方格有障碍物；

‘A’：表示该方格中有一个逆时针转向器；

‘C’：表示该方格中有一个顺时针转向器；

‘.’：表示该方格为空地。

【输出格式】

你的程序必须输出到标准输出。输出仅一个整数，表示最少需要推动的次数。若不能使所有机器人全部合并，输出-1。

【样例输入】

```
4 10 5
1.....
AA...x4...
..A..x....
2....x....
..C.3.A...
```

【样例输出】

5

【数据范围】

第一步：向右推动 3 号机器人，当它碰到转向器后会向上继续移动，直至碰到墙壁停止移动。第二步：向上推动 4 号机器人，当它碰到墙壁后停止移动，与 3 号机器人合并，构成 3-4 号机器人 第三步：向上推动 2 号机器人，当它碰到转向器后会向左移动，由于左侧为墙壁，故停留在原地。第四步：向右推动 2 号机器人，由于它在一个转向器上，故它会向上移动，直至碰到墙壁停止移动，与 1 号机器人合并，构成 1-2 号机器人。第五步：向左推动 3-4 号机器人，当它碰到墙壁后停止移动，与 1-2 号机器人合并，构成 1-4 号机器人。

$n \leq 9$, $w \leq 500$ 且 $h \leq 500$

Problem 2 money (money.cpp/c/pas)

【题目描述】

小蛇是金融部部长。最近她决定制造一系列新的货币。假设她要制造的货币的面值为 $x_1, x_2, x_3 \dots$ 那么 x_1 必须为 1, x_b 必须为 x_a 的正整数倍 ($b > a$)。例如 1, 5, 125, 250 就是一组合法的硬币序列，而 1,

5, 100, 125 就不是。不知从哪一天开始, 可爱的蛇爱上了一种萌物——兔纸! 从此, 小蛇便走上了遇上兔纸娃娃就买的不归路。某天, 小蛇看到了 N 只可爱的兔纸, 假设这 N 只兔纸的价钱分别是 $a_1, a_2 \dots a_N$ 。现在小蛇想知道, 在哪一组合法的硬币序列下, 买这 N 只兔纸所需要的硬币数最少。买兔纸时不能找零。

【输入格式】

第一行, 一个整数 N , 表示兔纸的个数

第二行, N 个用空格隔开的整数, 分别为 N 只兔纸的价钱

【输出格式】

一行, 一个整数, 表示最少付的钱币数。

【样例输入】

```
2
25 102
```

【样例输出】

```
4
```

【数据范围】

样例解释: 共有两只兔纸, 价钱分别为 25 和 102。现在小蛇构造 1, 25, 100 这样一组硬币序列, 那么付第一只兔纸只需要一个面值为 25 的硬币, 第二只兔纸需要一个面值为 100 的硬币和两个面值为 1 的硬币, 总共两只兔纸需要付 4 个硬币。这也是所有方案中最少所需要付的硬币数。 $1 \leq N \leq 50, 1 \leq a_i \leq 100,000$

Problem 3 概率充电器 (chance. cpp/c/pas)

【题目描述】

著名的电子产品品牌 SHOI 刚刚发布了引领世界潮流的下一代电子产品——概率充电器:

“采用全新纳米级加工技术,实现元件与导线能否通电完全由真随机数决定!SHOI 概率充电器,您生活不可或缺的必需品!能充上电吗?现在就试看看吧!

”SHOI 概率充电器由 $n-1$ 条导线连通了 n 个充电元件。进行充电时,每条导线是否可以导电以概率决定,每一个充电元件自身是否直接进行充电也由概率决定。

随后电能可以从直接充电的元件经过通电的导线使得其他充电元件进行间接充电。

作为 SHOI 公司的忠实客户,你无法抑制自己购买 SHOI 产品的冲动。在排了一个星期的长队之后终于入手了最新型号的 SHOI 概率充电器。

你迫不及待地将 SHOI 概率充电器插入电源——这时你突然想知道,进入充电状态的元件个数的期望是多少呢?

【输入格式】

第一行一个整数: n 。概率充电器的充电元件个数。充电元件由 $1-n$ 编号。

之后的 $n-1$ 行每行三个整数 a, b, p , 描述了一根导线连接了编号为 a 和 b 的

充电元件, 通电概率为 $p\%$ 。

第 $n+2$ 行 n 个整数: q_i 。表示 i 号元件直接充电的概率为 $q_i\%$ 。

【输出格式】

输出一行一个实数, 为进入充电状态的元件个数的期望, 四舍五入到六位小数。

【样例输入】

3

1 2 50

1 3 50

50 0 0

【样例输出】

1.000000

【数据范围】

对于 100% 的数据, $n \leq 500000, 0 \leq p, q_i \leq 100$ 。