

太戈编程练习题

(请选手务必仔细阅读本页内容)

一、题目概况

中文题目名称	长寿基因	可怜的蜗牛	月球脱险	魂器
英文题目与子目录名	longlife	snail	moon	steal
可执行文件名	longlife	snail	moon	steal
输入文件名	longlife.in	snail.in	moon.in	steal.in
输出文件名	longlife.out	snail.out	moon.out	steal.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
附加样例文件	无	无	无	无
结果比较方式	全文比较，过滤末行后空行，不过滤中间行行末空格			
题目类型	传统	传统	传统	传统
运行内存上限	256M	256M	256M	256M

二、提交源程序文件名

对于 C++语言	longlife.cpp	snail.cpp	moon.cpp	steal.cpp
----------	--------------	-----------	----------	-----------

三、编译命令（不包含任何优化开关）

对于 C++语言	<code>g++ -o longlife longlife.cpp -lm</code>	<code>g++ -o snail snail.cpp -lm</code>	<code>g++ -o moon moon.cpp -lm</code>	<code>g++ -o steal steal.cpp -lm</code>
----------	---	---	---------------------------------------	---

长寿基因

(longlife.cpp/c/pas)

时空限制：1s/256M，测试数据共 10 组

【问题描述】

众所周知，各种生物的遗传 DNA 都是由四种碱基组成，分别记为 A、G、C、T。据伟大的遗传学家卜靠谱教授数十年的潜心研究，生物的 DNA 序列中 TAT 序列出现次数越多，寿命越长。注意序列中包含的 TAT 可以重叠，例如 TATAT 算出现了两次 TAT。请你统计出现次数最多的样本名字及出现次数。

【输入格式】输入文件 **longlife.in**第一行一个整数 n ，表示样本数目。接下来有 $2n$ 行，每行一个由大写英文字母组成的字符串。这 $2n$ 行中，第 $2i-1$ ($1 \leq i \leq n$) 行的字符串表示第 i 个样本的名字，第 $2i$ 行的字符串表示第 i 个样本的碱基序列。**【输出格式】**输出文件 **longlife.out**

输出共两行，第一行是最“长寿”样本的名字。如果最长寿样本有并列多个，则按照他们的输入顺序将样本名字依次输出，相邻两个名字间用空格分隔。

第二行一个整数，表示最长寿样本中包含多少个 TAT。

【输入输出样例 1】

longlife.in	longlife.out
2 FDBST TATATGATCTAT JDSCD GCTATTATT	FDBST 3

【输入输出样例 2】

longlife.in	longlife.out
3 SHBND TATATATATATAT BJCPT TATTATTATTAT SZSPZ TATCGTATTATTATTATTAT	SHBND SZSPZ 6

【数据规模】10%数据， $n=1$ 100%数据， $n \leq 100$ ，样本名字长度不超过 20，样本序列长度不超过 200。

可怜的蜗牛

(snail.cpp/c/pas)

【问题描述】

Seldon 是个伟大的物理学家，他发明了瞬间传送装置，可以把一个不太重的物体从一个位置传送到另一个位置，甚至活的生命体也可以传送。不过一个传送装置只能使用一次。Seldon 决定用他的宠物蜗牛做实验。他把蜗牛放进院子的井里，蜗牛从井底以匀速向上爬行，速度为 v 米/小时。Seldon 在爬行路径上设置了 n 个传送装置，他可以遥控选择是否启用某个装置。第 i 个装置在距井底 a_i 米处，如果该装置被启用，当蜗牛第一次到达位置 a_i 时，会一瞬间回到井底（所用时间忽略不计，即可以认为蜗牛立刻从井底重新出发）。但蜗牛第二次爬到相同位置时，装置就不再起作用了。

例如 Seldon 启用了 $a_i=3$ 与 $a_i=5$ 的两个装置，蜗牛的速度 $v=1$ ，井的深度 $L=6$ ，则蜗牛的上爬过程如下：

- 用 3 小时爬到位置 3。
- 受 $a_i=3$ 的装置影响，回到了井底出发。
- 再用 3 小时爬到位置 3，然而因为是第二次到达， $a_i=3$ 的装置不起作用。
- 用 2 小时爬到位置 5。
- 受 $a_i=5$ 的装置影响，回到了井底出发。
- 用 6 小时爬到井口。花费的总时间为 14 小时。

现在，Seldon 有 q 个询问。对于第 i 个询问 t_i ，seldon 想知道他最少要启用多少个装置才能使蜗牛到达井口所用的时间大于 t_i 。

【输入格式】

输入文件 **snail.in**

第一行三个整数 n, L, v 分别表示装置的数量，井的深度，蜗牛的速度。

第二行 n 个整数。第 i 个整数 a_i 表示第 i 个装置的位置。（ $1 \leq i \leq n$ ）

第三行一个整数 q 表示询问个数。

接下来 q 行每行一个整数，第 i 行的整数 t_i 表示第 i 个询问。（ $1 \leq i \leq q$ ）

【输出格式】

输出文件 **snail.out**

输出 q 行，每行恰好一个整数，第 i 行的整数对应第 i 个询问的答案。（ $1 \leq i \leq q$ ）

如果 Seldon 无论如何都不能使蜗牛花费的时间大于 t_i ，请输出 -1。

【输入输出样例 1】

snail.in	snail.out
3 6 3	0
3 5 1	1
4	2
1	-1
3	
4	
5	

【样例 1 解释】

- 不启用任何装置，蜗牛需要 2 小时爬到井口。
- 启用装置 2，蜗牛需要 $11/3$ 小时爬到井口。（用时 $5/3$ 小时爬到装置 2 的位置并回到井底，再用 $6/3=2$ 小时爬到井口）
- 启用装置 1 和 2，蜗牛需要 $14/3$ 小时爬到井口。
- Seldon 无法使蜗牛用大于 5 小时的时间爬到井口。

【数据范围与提示】

对于测试点 1~8: $n=1$ 。

对于测试点 9~12: $n=2$ 。

对于测试点 13~17: $n, q \leq 1000$ 。

对于所有测试点: $1 \leq n, q \leq 2 \times 10^5$, $1 \leq v \leq L \leq 10^9$, $1 \leq a_i \leq L$, $1 \leq t_i \leq 10^9$ 。

数据保证 a_i 两两不同。

月球脱险

(moon.cpp/c/pas)

时空限制: 1s/256M, 测试数据共 10 组

【问题描述】

月球 1 号基地发生爆炸, 宇航员要逃到 2 号基地。两地之间距离为 L , 宇航服能加载的氧气量最大为 C 升, 每耗费 1 升氧气宇航员能够行走的距离为 U 。途中有 N 个氧气站可以补充氧气 ($N \geq 0$), 第 i 个氧气站 ($1 \leq i \leq N$) 离 1 号基地的距离为 D_i , 补充 1 升氧气需要耗费的宇航服电力为 P_i 。出发时宇航服中氧气量为 0, 出发点补充 1 升氧气 耗费宇航服电力 P 。求宇航员到达 2 号基地最少需要耗费多少宇航服电力? 计算结果四舍五入至小数点后两位。如果无法到达目的地, 则输出 -1。

【输入格式】

输入文件名为 moon.in。

输入文件第一行是四个浮点数 L 、 C 、 U 、 P 和 1 个整数 N , 每两个数据用一个空格隔开。接下来 n 行, 每行 2 个浮点数 D_i 和 P_i , 中间用一个空格隔开。

【输出格式】

输出文件名为 moon.out。

输出文件包含 1 个浮点数 (两位小数) 或者 -1。

【输入输出样例 1】

moon.in	moon.out
300 10 20 2 2 100 3 200 1	25.00

【输入输出样例 1 说明】

每升氧气行走距离为 20。出发时加满氧气 10 升, 耗费电力 20, 正好可以走到距离 200 处的第 2 个氧气站, 再加 5 升氧气, 耗费电力 5, 即可到达终点。一共耗费 25 电力。

【数据规模与约定】对于 100% 的数据, $0 \leq N \leq 100$ 。

魂器

(steal.cpp/c/pas)

时空限制：1s/256M，测试数据共 20 组

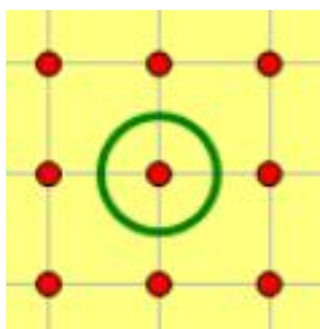
【问题描述】

哈利波特来到古灵阁，试图偷取魂器。

古灵阁的地图可以描述为一个 n 行 m 列的矩形。在这里，我们规定每一行中从左到右为 x 轴正方向，每一列中从下到上为 y 轴正方向，左下角的点坐标为 $(1, 1)$ 。矩形中的点可以分为 4 种：

1. 起点，也就是哈利波特的所在点，在地图中用字符 S 代表。
2. 终点，也就是魂器的所在点，在地图中用字符 T 代表。
3. 卫兵，在地图中用一个正整数 $a_{i,j}$ 代表。在这里，一个卫兵 (i, j) 可以观察到与他曼哈顿距离小于 $a_{i,j}$ 的点。也就是卫兵 (i, j) 可以观察到所有满足 $|x-i| + |y-j| < a_{i,j}$ 的点 (x, y) 。
4. 空地，在地图中用字符 $.$ 代表。

哈利波特的正常移动方式为每秒向八连通的任意方向前进一格。如下图，中间的点为哈利波特当前所在点，每一秒，他可以走向其余的八个点。



需要注意的是，正常移动时，哈利波特不能踏进任何一个有卫兵或者卫兵能观察到的格子。当然，他也不能走出古灵阁，也就是说，无论何时，哈利波特的坐标 (x, y) 都必须满足 $1 \leq x \leq m$ 且 $1 \leq y \leq n$ 。

哈利波特还有两种技能：隐身和瞬移。

1. 隐身：下一秒哈利波特披上隐身衣进入隐身状态，卫兵观察不到哈利波特，哈利波特可以进入卫兵的观察范围内，但仍然不能进入卫兵所在的格子。注意这个状态只能维持一秒。
2. 瞬移：哈利波特带着最新版飞天扫帚“光轮 30 万”，骑上它下一秒移动的距离改为 d ，但这时只能向上下左右四个方向移动。即可以移动到 $(x+d, y)$, $(x-d, y)$, $(x, y+d)$, $(x, y-d)$ 。在本题中，两种技能可以同时使用，而且不考虑冷却时间，即一次用完可以立即用下一次，两种技能都分别有使用次数限制，你也可以不用完所有次数。

现在给出古灵阁的地图，请计算哈利波特到达魂器所在点所需的最短时间。此外，在所用时间相同情况下，哈利波特希望使用的两种技能总次数尽可能少；在所用时间与技能次数相同情况下，哈利波特希望使用的隐身次数尽可能少。

【输入格式】

输入文件 **steal.in**

第一行五个整数 $n, m, c1, c2, d$, 代表地图的大小为 $n \times m$, 隐身的使用限制次数为 $c1$, 瞬移的使用限制次数为 $c2$ 和一次瞬移的距离为 d 。

接下来 n 行, 每行 m 个元素。每个元素为字符 $S, T, .$ 或者一个正整数 $a_{i,j}$ 代表一个格点, 具体含义详见题目描述

【输出格式】

输出文件 **steal.out**

若哈利波特无法到达魂器所在点, 则输出一行一个-1。

否则输出一行三个整数 $t, u1, u2$, 依次代表所需的最短时间, 隐身的使用次数与瞬移的使用次数。

【输入输出样例 1】

steal.in	steal.out
5 4 0 0 5 . 1 T 1 . . . 2 . 1 . . S . . . 1 . . .	3 0 0

【样例 1 解释】

起点为(1, 2), 哈利波特可以依次走到(1, 3), (2, 4), (3, 5)到达终点。

【输入输出样例 2】

steal.in	steal.out
8 6 2 3 3 . S . 2 . 2 . 2 . . . 1 . T . 3 . 1 . . 3	3 1 3

【样例 2 解释】

起点为(2, 8), 哈利波特可以依次走到(5, 8), (5, 5), (5, 2)。

【输入输出样例 3】

steal.in	steal.out
----------	-----------

8 6 5 5 2 . S 1 1 3 2 . 1 2 3 2 2 1 3 3 2 4 1 4 3 2 6 1 5 T 2 8 1 6 3 2 10	-1
--	----

【数据范围与提示】

- 对于测试点 1~ 6: $n, m \leq 10, c1=c2=0$, 保证所需的最短时间不超过 5 或者无解。
- 对于测试点 7~ 10: $n, m \leq 20, c1=c2=0$, 保证 T 的位置不在任何一个卫兵的观察范围之内。
- 对于测试点 11~ 12: $n, m \leq 20, c1=0$
- 对于测试点 13~ 14: $n, m \leq 20, c1, c2 \leq 5$ 。
- 对于测试点 15~ 16: 卫兵个数不超过 350。
- 对于所有测试点: $2 \leq n, m \leq 350, 1 \leq ai, j \leq 350, 0 \leq c1, c2 \leq 15, 1 \leq d \leq 350$ 。
- 保证 S 的位置不在任何卫兵的观察范围中。