

期末测试

测试时间 2022年6月12日 8: 00 ~ 12: 00

题目	消灭怪物	单词表	阴阳项链	交通网络	激光防护
文件名	monster.cpp	wordlist.cpp	yy.cpp	network.cpp	laser.cpp
输入输出	monster.in/.out	wordlist.in/.out	yy.in/.out	network.in/.out	laser.in/.out
时间限制	2s	1s	1s	2s	1s
空间限制	256MB	32MB	256MB	256MB	256MB
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: 消灭怪物(monster)

文件名 `monster.cpp/.in/.out` , 时间限制2秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

问题描述

$n \times m$ 的网格中有若干只怪物, 有怪的地方用 `*` 符号表示, 没有怪物的地方用 `.` 符号表示。

为了消灭所有的怪物, 你决定使用技能 **十字斩击**。

使用 **十字斩击** 需要选择某个地块释放, 消灭与选择地块同行以及同列的所有怪物。

使用技能的代价不小, 你希望计算出消灭所有怪物所需的最少使用次数。

输入格式

本题有多组测试数据, 第一行输入数据组数 T , 对每组数据:

第一行有 2 个正整数 n, m 。

接下来 n 行, 每行有长度为 m 的字符串, 仅包含 `*` 与 `.`, 描述网格中的怪物信息。

输出格式

对每组测试数据, 输出 1 个整数, 表示技能的最少使用次数。

样例1

样例输入

```
1
8 10
.....
..**.*.*
.*.....*
.*.....*
.*.....*
.*.....*
.....*****
.....
.....*
```

样例输出

```
2
```

样例2

见下发文件 `monster2.in/.out`

数据范围

对于 30% 的数据, $1 \leq n, m \leq 8$

对于 60% 的数据, $1 \leq n, m \leq 18$

对于 100% 的数据, $1 \leq T \leq 5, 1 \leq n, m \leq 24$

T2: 单词表(wordlist)

文件名 wordlist.cpp/.in/.out ，时间限制1秒，空间限制32MB，编译开启 -O2 优化，支持 C++14。

问题描述

背单词是学习英语中必不可少的一环。这周你计划要背 n 个单词，你将它们按字典序排好，准备填入 Excel 表格中以便查询。

你填写的规则是，按列的顺序从上到下填写单词，当一列填满后再开始填写下一列。Excel 表格会根据每一列单词的长度动态调整当前列的宽度，每一列的宽度与这列中最长的单词保持一致。同时，为了分隔两列的单词，两列之间会留有一个字符长度的空白。如下表所示

ability	advice
able	aggressive
absence	agriculture
admission	aim
advertisement	

你的电脑屏幕的宽度有限，一行只能显示 w 个字符。如果表格的宽度超过了电脑屏幕，就会出现水平滑块，不能用鼠标滚轮操控，十分不便。

与此同时，虽然竖直方向的滑块可以直接用滚轮滑动，但你也不希望表格太长，这不优雅。你想知道在单词表的宽度不超过电脑屏幕的前提下，高度最少为多少个字符？

输入格式

第一行输入 2 个整数 n 、 w 。

第二行输入 n 个整数 a_i ，表示字典序第 i 的单词的长度。

输出格式

输出一个整数，表示单词表高度的最小值。

样例1

样例输入

```
6 10
1 1 4 5 1 4
```

样例输出

```
3
```

样例解释

高度为 3 时，前 3 个单词在第 1 列，后 3 个单词在第 2 列。

此时第 1 列的宽度是 4，第 2 列的宽度是 5，两列之间还有宽度为 1 的空白，总宽度为 10，恰好符合宽度限制。

样例2

样例输入

```
10 54
13 13 10 17 2 3 17 20 12 2
```

样例输出

```
4
```

样例解释

高度为 4 时，第 1 列包含第 1, 2, 3, 4 个单词，宽度为 17；第 2 列包含第 5, 6, 7, 8 个单词，宽度为 20；第 3 列包含第 9, 10 个单词，宽度为 12。总宽度为 $17 + 1 + 20 + 1 + 12 = 51$ ，符合宽度限制。

样例3

见下发文件 wordlist3.in/.out

数据范围

对于 30% 的数据， $1 \leq n \leq 1000$

对于 60% 的数据， $1 \leq n \leq 10^5$

对于 100% 的数据， $1 \leq n \leq 10^6$ ， $1 \leq a_i \leq w \leq 10^9$

其中前 80% 的测试点内存限制为 256MB，后 20% 的测试点减少到 32MB。

提示

本题输入量较大，可以考虑使用 fread 进行数据读入，下面给出参考程序。

```
#include <stdio>
#include <ctype>
const int SIZE = 1 << 20;
char buf[SIZE], *p1, *p2;
#define gc (p1==p2&&(p2=(p1=buf)+fread(buf,1,SIZE,stdin),p1==p2)?EOF:*p1++)
void read(int &x) {
    x = 0;
    char ch;
    while(!isdigit(ch = gc));
    do {
        x = x*10 + ch - '0';
    }while(isdigit(ch = gc));
}
```

T3: 阴阳项链(yy)

文件名 `yy.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

你有 $2n$ 颗宝珠, 其中 n 颗具有阴性能量, 另外 n 颗具有阳性能量, 分别从 $1 \sim n$ 编号。

你要把它们串成环构成一串阴阳项链, 保证阴性宝珠和阳性宝珠间隔排列。

进一步研究发现, 有些阴性宝珠和阳性宝珠之间会发生互斥, 你总结出了共 m 条这样的互斥关系: x_i 号阳性宝珠若与 y_i 号阴性宝珠相邻, 则会失去能量。

合理排列阴阳宝珠的顺序, 使得项链中失去能量的阳性宝珠数量最少。

输入格式

本题有多组测试数据, 第一行输入数据组数 T , 对每组数据:

第一行输入 2 个整数 n, m 。

接下来 m 行, 每行输入 2 个整数 x_i, y_i 。

输出格式

对每组数据, 输出一个整数, 表示失去能量的阳性宝珠数量的最小值。

样例1

样例输入

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

样例输出

```
1
1
```

样例2

见下发文件 `yy2.in/.out`

样例3

见下发文件 `yy3.in/.out`

数据范围

对于 30% 的数据, $1 \leq n \leq 5$

对于 60% 的数据, $1 \leq n \leq 8$

对于 100% 的数据, $1 \leq T \leq 20, 1 \leq n \leq 9, 0 \leq m \leq n^2, 1 \leq x_i, y_i \leq n$

T4: 交通网络(network)

文件名 `network.cpp/.in/.out` , 时间限制2秒, 空间限制256MB, 编译开启 `-O2` 优化, 支持 C++14。

问题描述

某城市的交通网络由 n 个站点和 m 条路径构成, 路径有的可以双向通行, 有的却是单行道。站点编号为 $1 \sim n$ 。

该城的所有站点和路径均在 $A \times B$ 的矩形内, 左下角坐标为 $(0, 0)$, 右上角坐标为 (A, B) , 第 i 个站点的坐标为 (x_i, y_i) 。

为保证交通顺畅, 该城市的任意两条路径都不会在非端点处相交。

现在某运输队要从西城的某些站点运送货物到东城。其中, 西城特指 $x_i = 0$ 的站点, 东城则是 $x_i = A$ 的站点。

请你帮忙求出, 从每一个西城的站点出发, 能到达多少个东城站点。

输入格式

第一行输入 4 个正整数 n, m, A, B 。

接下来 n 行, 每行有 2 个整数 x_i, y_i , 描述第 i 个站点的坐标。

再接下来 m 行, 每行有 3 个整数 u_i, v_i, t_i , 描述一条连接站点 u_i 与 v_i 的路径。当 $t_i = 1$ 时, 这是由 u_i 通往 v_i 的单行道; 当 $t_i = 2$ 时, 该路径可双向通行。

输出格式

对每一个西城站点 ($x_i = 0$), 输出从它出发能到达多少个东城站点。

按 y_i **从大到小** 的顺序输出, 每个答案独占一行。

样例1

样例输入

```
5 3 1 3
0 0
0 1
0 2
1 0
1 1
1 4 1
1 5 2
3 5 2
```

样例输出

2
0
2

样例2

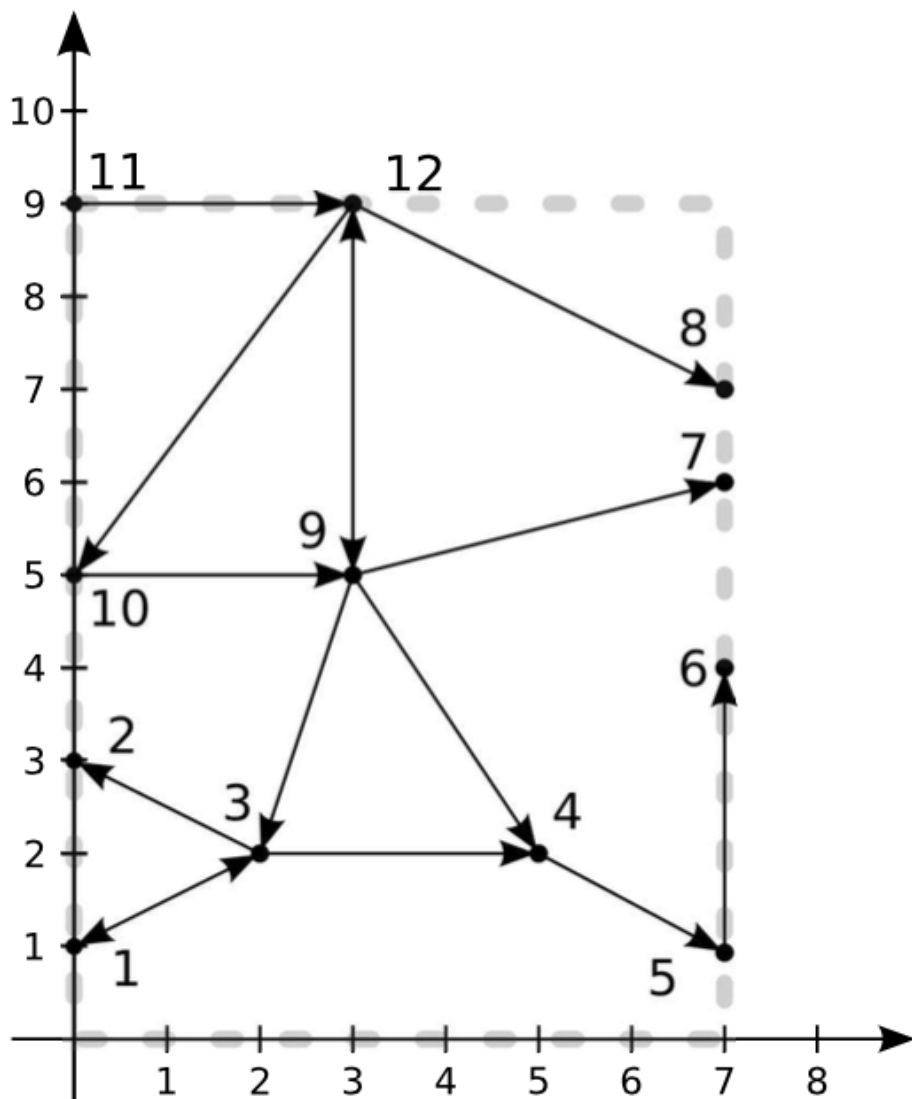
样例输入

12 13 7 9
0 1
0 3
2 2
5 2
7 1
7 4
7 6
7 7
3 5
0 5
0 9
3 9
1 3 2
3 2 1
3 4 1
4 5 1
5 6 1
9 3 1
9 4 1
9 7 1
9 12 2
10 9 1
11 12 1
12 8 1
12 10 1

样例输出

4
4
0
2

样例解释



样例3

见下发文件 `network3.in/.out`

数据范围

对于 30% 的数据, $1 \leq n \leq 5000$, $1 \leq m \leq 10000$

对于 100% 的数据, $1 \leq n \leq 300000$, $1 \leq m \leq 900000$, $1 \leq A, B \leq 10^9$, $0 \leq x_i \leq A$, $0 \leq y_i \leq B$, $1 \leq u_i, v_i \leq n$, $t_i \in \{1, 2\}$, 数据保证无重边, 每个站点坐标唯一, 且至少有一个西城点

本题输入量较大, 可以考虑使用 `fread` 进行数据读入。

T5: 激光防护(laser)

文件名 laser.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。

本题为 **spj** 评测。

问题描述

某军事要地正在安装激光防护系统，以确保每一个空白地块均被激光打击覆盖。

该要地由 $n \times m$ 的地块构成，其中 `.` 表示空白地块，`s` 表示激光发射器，`#` 表示墙体等障碍物。

所有激光发射器的位置均已确定。由于改变发射方向过于麻烦，现在要固定每个发射器的发射方向。激光发射器的发射方向有左右或上下，每次会 **同时** 发射 2 束激光射线，打击该方向的所有空白地块 `.`。当激光遇到障碍物 `#` 时会立即消失。

当激光遇到激光发射器 `s` 时，强大的能量会摧毁该发射器，造成巨大的损失，因此在决定发射方向时要多加小心。

你的任务是，为每一个激光发射器确定方向，确保每一个空白地块 `.` 均被激光打击覆盖，且不存在某个激光发射器发射的激光能摧毁任意一个激光发射器（包括它本身）。

输入格式

本题有多组测试数据，第一行输入数据组数 T ，对每组数据：

第一行输入 2 个整数 n, m 。

接下来 n 行，每行输入一个长度为 m 的字符串，字符串仅包含题面中提到的 `.``s` 三种字符。

输出格式

对每组数据，若存在满足题意的方案，输出 `Yes`，并输出对应的方案；若不存在，输出 `No`。

输出方案时，使用 `-` 与 `|` 替换原本表示激光发射器的 `s`，`-` 表示发射方向为左右，`|` 表示发射方向为上下。其余字符保持原样输出。具体可参照样例。

样例1

样例输入

3
1 3
S.S
3 4
#.##
#SS#

2 2
S.
#S

样例输出

No
Yes
#.##
#||#

Yes
-.
#-

样例2

见下发文件 laser2.in/.out

数据范围

- 对于 20% 的数据, $1 \leq n \times m \leq 5$
- 对于 80% 的数据, $1 \leq T \leq 20, 1 \leq n \times m \leq 250$
- 对于 100% 的数据, $1 \leq n, m \leq 500$, 单个输入文件大小不超过 1MB