题目翻译

1. 首都：

内存：537MB 时限：2.5s 题型：传统

/usr/bin/g++ -DEVAL -std=gnu++14 -O2 -pipe -static -s -o capital\_city capital\_city.cpp

题目描述：

JOI王国有N个城镇，从1到N，有N-1条道路连接城镇。第i条（1≤i≤N 1）连接城镇Ai和Bi。任何一条路都可以双向通过。你可以通过一些道路从任何城镇到任何其他城镇旅行。

目前，JOI王国分为K个城市，从1到K，城镇j（1≤j≤N）属于城市Cj. 每个城市至少有一个城镇。

K总统是JOI王国的国王。他将选择一个城市作为首都。出于安全考虑，首都必须满足以下条件：

你可以从首都的任意城镇出发只经过属于首都的城市来前往首都的其他所有城镇

但是，K总统注意到，可能没有一个城市满足这个条件，他无法选择首都。

为了解决这个问题，K总统将合并城市。准确地说，他可以做下面的操作：

选择满足1≤x≤K，1≤y≤K并且x!=y的x和y，改变属于城市y的城镇使得所有属于城市y的城镇成为属于城市x的城镇。

由于合并城市的成本太高，K总统希望尽量减少城市合并的次数来选择出一个城市作为首都。

写一个程序，给出JOI王国的各城市的城镇和道路的结构和各城镇当前所属，计算合并城市的最小数目。

题目输入：

所有输入为整数

N K

A1 B1

....

A{N-1} B{N-1}

C1

...

CN

题目输出：

输出为了能够选择一个城市作为首都的最小合并次数。

数据限制：

• 1 ≤ N ≤ 200 000.

• 1 ≤ K ≤ N.

• 1 ≤ Ai ≤ N (1 ≤ i ≤ N-1).

• 1 ≤ Bi ≤ N (1 ≤ i ≤ N-1).

• 你可以从任意城镇通过一些道路到达其他任意城镇

• 1 ≤ Cj ≤ K (1 ≤ j ≤ N).

• 对于每一个k (1 ≤ k ≤ K), 存在一个整数j (1 ≤ j ≤ N) 使得 Cj = k.

子任务：

1. (1 point) N ≤ 20.

2. (10 points) N ≤ 2 000.

3. (30 points) 每座城镇最多和两座城镇通过道路直接相连

4. (59 points) 无额外限制

样例一：

输入：

6 3

2 1

3 5

6 2

3 4

2 3

1

3

1

2

3

2

输出：

1

解释：

为了合并城市3和城市1，选择(x, y) = (1, 3). 此时城市1可以作为首都。

初始时你不能选择城市作为首都，因此最小合并次数为 1.

样例一输入满足子任务1、2、4的限制

样例二：

输入：

8 4

4 1

1 3

3 6

6 7

7 2

2 5

5 8

2

4

3

1

1

2

3

4

输出：

1

解释：

样例二输入满足子任务1、2、3、4的限制

样例三：

输入：

12 4

7 9

1 3

4 6

2 4

10 12

1 2

2 10

11 1

2 8

5 3

6 7

3

1

1

2

4

3

3

2

2

3

4

4

输出：

2

解释：

样例三输入满足子任务1、2、4的限制

1. 传说中的饭团制作者

类型：提答题

题目描述：

你是一个制作美味饭团dango的专业人士。现在你再用竹签串dango。

dango位于由R行和C列组成的矩形网格中。每个单元格包含一个dango。dango的颜色是粉色（P）、白色（W）或绿色（G）。您将在垂直方向（从上到下）、水平方向（从左到右）或倾斜方向（从左上到右下或从右上到左下）上选择三个连续的dango。然后你要从这些格子的一端到另一端拿出这三个Dango，并把它们按顺序串成一串，这样就可以得到一串dango。例如，如果在垂直方向上选择三个连续的dango，可以以上中下或下中上的格获子顺序取出dango，并按照这个顺序，把它们串在一根棍子上。你不能按中底上或底上中的顺序取出dango。而且，你不能把一个dango串到一根以上的竹签里。

如果一串dango的颜色是粉-白-绿或绿-白-粉，那么这串dango是漂亮的。你想做尽可能多的串出漂亮的dango。

你最多能做出多少串漂亮的dango？

题目输入：

总共6组输入，每组形式如下

R C

D1

...

DR

其中Di(1 ≤ i ≤ R)是一个长度为C的、由字符P、W、G组成的字符串。Di的第j个元素(1 ≤ j ≤ C)是从上往下数第i行、从左往右数第j列的格子中的dango颜色。

题目输出：

输出格式如下：

S1

...

SR

这里S i（1≤i≤R）是一个长度C的字符串，由P、W、G、|、-、\和/组成。Si的第j个字符（1≤j≤C）描述了位于从上往下第i行、从左往右第j列的dango串起来的方向。

• Si的第j个字符为 | 当且仅当你使用了那个格子的dango以及它上下相邻的dango串成了一串美丽的dango串

• Si的第j个字符为 - 当且仅当你使用了那个格子的dango以及它左右相邻的dango串成了一串美丽的dango串

• Si的第j个字符为 \ 当且仅当你使用了那个格子的dango以及它左上、右下相邻的dango串成了一串美丽的dango串

• Si的第j个字符为 / 当且仅当你使用了那个格子的dango以及它左下、右上相邻的dango串成了一串美丽的dango串

• 否则，Si的第j个字符会根据那个格子dango的颜色显示为 P, W, G, 即它和Di的第j个字符相同

提交要求：

为每组输入数据提交输出

数据限制：

• 3 ≤ R ≤ 500.

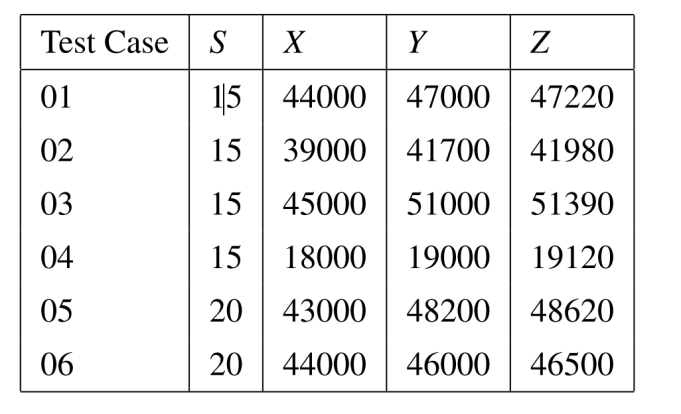
• 3 ≤ C ≤ 500.

• Di (1 ≤ i ≤ R) is a string of length C consisting of P, W, and G.

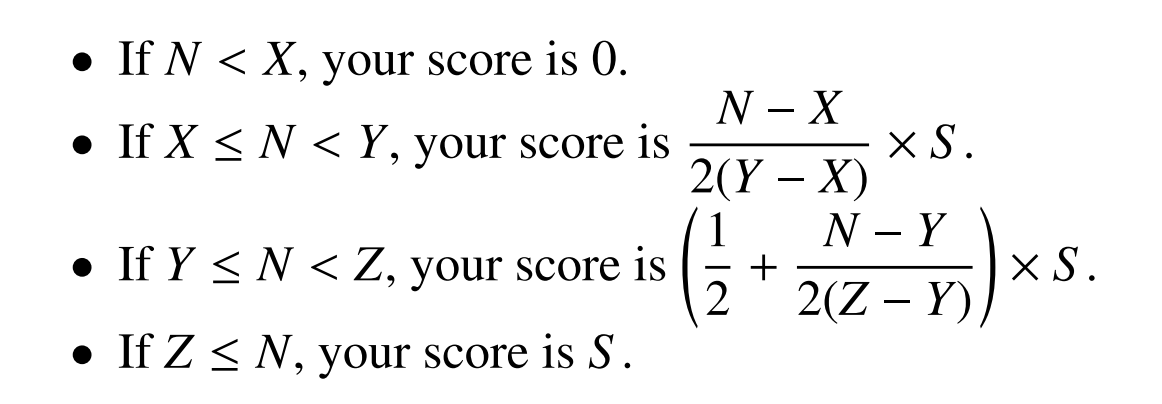
得分设置：

此题的分数由如下方式计算。

对于每个测试点，我们定义4个值S（表示分配的分数）、X、Y、Z



对于每组测试点，设N是你提交的答案能够串出的美丽dango串的数量，你的分数计算如下：



此题的总分数是各测试用例的分数之和，四舍五入为最接近的整数。

但是，如果你的输出数据不正确，那么你的测试用例得分是0。例如，根据你输出数据中的字符|、-、\、/不能做出漂亮的木棒，数据与输入数据冲突，或者输出数据的格式错误。

样例一：

输入：

3 4

PWGP

WGPW

GWPG

输出：

P-GP

WGP|

G-PG

解释：

此样例中，你做出了三串美丽的dango。

注意WGP不是一个可行的顺序

样例二：

输入：

3 4

PWWP

WWWW

PGGP

输出：

PWWP

W\/W

PGGP

解释：

可以串出两个美丽的dango串

可视化：

有一个可视化工具可用于此题，它可以可视化地显示输入数据或输出数据。您可以从竞赛系统下载可视化工具作为此题的附件文件。

若要使用visualizer，应通过web浏览器打开附加的visualizer.html，然后选择或拖动文件到该网页中。请注意，可视化工具没有完全检查文件格式的正确性。如果输入文件的格式错误，则可能无法正确执行。而且，如果R或C的值超过约束，它可能也不会被执行

1. 治疗计划

内存：537MB 时限：2s 题型：传统题

/usr/bin/g++ -DEVAL -std=gnu++14 -O2 -pipe -static -s -o treatment treatment.cpp

题目描述：

JOI王国有N座房子，从1到N编号。房子按数字顺序排列在一条线上。每户人家都住一个居民。居住在房屋x（1≤x≤N）中的居民由x表示。

最近出现了新的病毒，每个居民都感染了这种病毒。为了解决这个问题，提出了M个治疗计划。计划成本i（1≤i≤M）为Ci. 如果执行了计划i，会发生以下情况：

在第Ti天的晚上，如果居民x（Li≤x≤Ri）被病毒感染了，那么x会被治好。

这种病毒通过邻近的居民以如下方式传播。

如果某一天早上居民x（1≤x≤N）感染了病毒，那么公民x-1（如果x≥2）和公民x+1（如果x≤N-1）在同一天中午感染病毒。

已经接受治疗的公民可能再次感染病毒。

既然你是JOI王国的部长，你需要选择一些项目，以便以下条件满足。

条件：当所有被选择的治疗计划被执行之后，没有居民感染病毒

有可能一天实行多个治疗计划。

编写一个程序，根据房屋数量和治疗计划的信息，确定是否可以满足上述条件，并在可能的情况下计算最小总成本。

题目输入：

所有输入为整数：

N M

T1 L1 R1 C1

...

TM LM RM CM

题目输出：

如果条件不能被满足，输出-1。否则，输出最小的总成本。

数据限制：

• 1 ≤ N ≤ 1 000 000 000.

• 1 ≤ M ≤ 100 000.

• 1 ≤ Ti ≤ 1 000 000 000 (1 ≤ i ≤ M).

• 1 ≤ Li ≤ Ri ≤ N (1 ≤ i ≤ M).

• 1 ≤ Ci ≤ 1 000 000 000 (1 ≤ i ≤ M).

子任务：

1. (4 points) Ti = 1 (1 ≤ i ≤ M).

2. (5 points) M ≤ 16.

3. (30 points) M ≤ 5 000.

4. (61 points) No additional constraints.

样例一：

输入：

10 5

2 5 10 3

1 1 6 5

5 2 8 3

7 6 10 4

4 1 3 1

输出：

7

解释：

你可以按如下方式执行计划：

• 第二天的晚上，执行计划1，居民 5, 6, 7, 8, 9, 10被治好. 居民 1, 2, 3, 4 仍被感染

• 第三天中午，居民 5 被感染，此时居民 1, 2, 3, 4, 5 仍被感染

• 第四天中午，居民 6 被感染. 此时居民1, 2, 3, 4, 5, 6 被感染.

• 第四天晚上，执行计划5，居民1, 2, 3被治好。 此时居民4, 5, 6 仍被感染.

• 第五天中午，居民 3 和 7被感染.此时居民 3, 4, 5, 6, 7 被感染

• 第五天晚上，执行计划3，居民 3, 4, 5, 6, 7 被治好. 之后，无居民感染病毒。

执行计划1、3、5的总花费为7. 因为不可能有比7更小的花费，所以输出7.

样例二：

输入：

10 5

2 6 10 3

1 1 5 5

5 2 7 3

8 6 10 4

4 1 3 1

输出：

-1

解释：

条件不能被满足，输出-1

样例三

输入：

10 5

1 5 10 4

1 1 6 5

1 4 8 3

1 6 10 3

1 1 3 1

输出：

7

解释：

此样例满足子任务1的限制