

第 40 届全国信息学奥林匹克竞赛

CCF NOI2023 模拟题

第一试

竞赛时间：7:30–12:20

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	最大值	染色	剖分
英文题目与子目录名	max	paint	decompose
可执行文件名	max	paint	decompose
输入文件名	max.in	paint.in	decompose.in
输出文件名	max.out	paint.out	decompose.out
每个测试点时限	2.0 秒	1.0 秒	1.0 秒
测试点数目	10	20	20
测试点是否等分	是	是	是
附加样例文件	有	有	有
结果比较方式	全文比较 (过滤行末空格及文末回车)		
题目类型	传统型	传统型	传统型
运行内存上限	512MB	512MB	512MB

二. 提交源程序文件名

对于 C++ 语言	max.cpp	paint.cpp	decompose.cpp
-----------	---------	-----------	---------------

三. 编译选项

对于 C++ 语言	-lm -std=c++14 -O2
-----------	--------------------

四. 注意事项:

- 文件名 (程序名和输入输出文件名) 必须使用英文小写。
- C/C++ 中函数 `main()` 的返回值类型必须是 `int`, 程序正常结束时的返回值必须是 `0`。
- 全国统一评测时采用的机器配置为: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz, 内存 8GB。上述时限以此配置为准。
- 只提供 Linux 格式附加样例文件。
- 特别提醒: 评测在当前最新公布的 NOI Linux 下进行, 各语言的编译器版本以其为准。

1. 最大值(max)

1.1 Description

一个长为 n 的序列 A ，从1开始标号，一开始全为0，现在小C想对它进行 m 次操作。

对第 i 次操作，他会选定恰好一个二元组 $(j, k), j \in [1, n], k \in [0, c]$ ，并令 $A_j = A_j + k$ ，其中选中二元组 (j, k) 的概率为 $P_{i,j,k}$ 。

小C本来是想问你区间最大值的历史版本和的历史最大值的期望的，但鉴于这是一道签到题，现在他只想知道 m 次操作后整个序列最大值的期望，对 $10^9 + 7$ 取模。

1.2 Input

从文件`max.in`中读入数据。

第一行三个整数 n, m, c ，含义见问题描述。

接下来 m 个 n 行 $c + 1$ 列的矩阵，第 i 个矩阵的第 j 行第 k 个元素表示 $P_{i,j,k-1}$ 的值，方便起见，这里给出的概率是模意义下的值。

1.3 Output

输出到文件`max.out`中。

输出一个整数表示答案。

1.4 Sample1

1.4.1 Input

```
3 1 1
3425 734783767
2345 34674684
980733421 249802373
```

1.4.2 Output

```
19260817
```

1.5 Sample2

见选手目录下的 *max/max2.in* 与 *max/max2.ans*.

1.6 Sample3

见选手目录下的 *max/max3.in* 与 *max/max3.ans*.

1.7 Subtasks

对于所有数据, 有 $1 \leq n \leq 40, 1 \leq m \leq 10, 1 \leq c \leq 3$.

$\forall i, \sum_{j \in [1, n], k \in [0, c]} P_{i, j, k} \equiv 1 \pmod{10^9 + 7}, 0 \leq P_{i, j, k} < 10^9 + 7$.

- subtask1(17%), $n, m = 5$.
- subtask2(14%), $n = 2$.
- subtask3(14%), $n = 5$.
- subtask4(34%), $c = 1, P_{i, j, 0} = 0$.
- subtask5(21%), 没有特殊的约定.

每个subtask只有一组数据.

2. 染色 (paint)

2.1 Description

小C很喜欢二维染色问题，这天他拿来了一个 $w \times h$ 的二维平面，初始时均为白色。然后他在上面设置了 n 个关键点 (X_i, Y_i) ，对于每个关键点他会选择进行下列操作的一个：

- 将 $x > X_i$ 的部分染成黑色。
- 将 $x < X_i$ 的部分染成黑色。
- 将 $y > Y_i$ 的部分染成黑色。
- 将 $y < Y_i$ 的部分染成黑色。

(图示参见样例解释)

他本来是想让你支持单点修改以及可持久化然后把空间限制开成 $1M$ 的，但鉴于这只是第二题，现在他只想最大化所有操作结束之后白色部分的**周长**(不难发现白色部分一定是个矩形)。特别地，如果没有白色部分，设其周长为0。

2.2 Input

从文件`paint.in`中读入数据。

第一行三个整数 w, h, n 。

接下来 n 行每行两个数表示 X_i, Y_i 。

2.3 Output

输出到文件`paint.out`中。

一行一个整数表示最大周长。

2.4 Sample1

2.4.1 Input

10 10 4

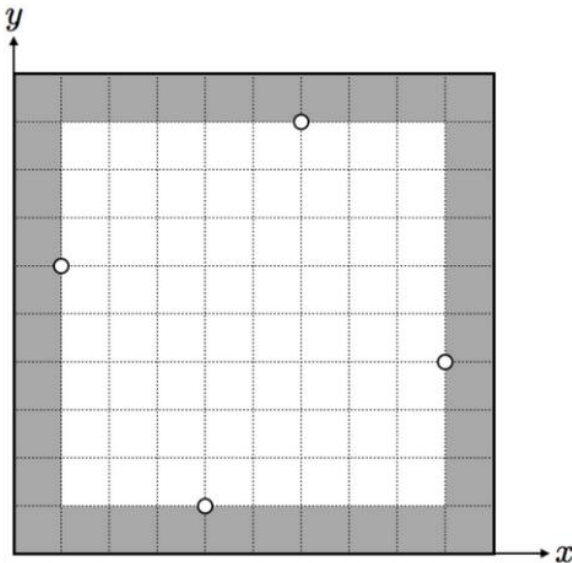
1 6
4 1
6 9
9 4

2.4.2 Output

32

2.4.3 Explanation

最优解如下图所示：



2.5 Sample2

2.5.1 Input

10 10 4
2 2
4 4
7 7
9 9

2.5.2 Output

26

2.5.3 Explanation

注意平面边界也可以成为最后得到的矩形的边界。

2.6 Sample3

见选手目录下的 *paint/paint3.in* 与 *paint/paint3.ans*。

2.7 Subtasks

对于所有数据，有 $1 \leq w, h \leq 10^8, 0 \leq n \leq 2 \times 10^5, 0 \leq X_i \leq w, 0 \leq Y_i \leq h$ 。

- subtask1(17%), $n \leq 10$.
- subtask2(16%), $n \leq 100$.
- subtask3(15%), $n \leq 1500$.
- subtask4(14%), $n \leq 7000$.
- subtask5(38%), $n \leq 2 \times 10^5$.

3. 剖分(decompose)

3.1 Description

小C有一棵 n 个点的树，1号点为根，每个点有 L 个权值，表示为 $w[u][i], u \in [1, n], i \in [1, L]$.

现在他想对这棵树进行树链剖分，于是fatesky教给他一种自创的剖分方法。具体地，一棵树的剖分可以表示为若干条链 S_1, S_2, \dots, S_k ，满足：

- 每个点属于且仅属于一条链。
- 一条链在树上是一个连通块，即对 $\forall i, u, v \in S_i$ ，从 u 到 v 的简单路径不经过任何不在 S_i 中的节点。
- $\forall i, S_i$ 的长度不超过 L 。
- 链中所有节点深度不同。

设一条链按深度从大到小排序后为 u_1, u_2, \dots, u_m ，fatesky定义一条链的权值为 $\sum_{i=1}^m w[u_i][i]$ ，一种剖分的权值为所有链的权值和。现在他想最大化剖分的权值。

小C本来是想让你支持链修改，子树查询，以及Link, Cut操作的，但考虑到这不是CTSC模拟题，现在他只需要你支持单点的权值修改。具体地，他会给出 q 个修改操作，每个修改操作给出一个点 u 以及 L 个值，表示修改之后的 $w[u][i]$ 。每个修改操作之后，你需要回答最大的剖分权值。

3.2 Input

从文件`decompose.in`中读入数据。

第一行三个整数 n, q, L 。

接下来一行 $n-1$ 个整数，第 i 个为 f_{i+1} ，表示树上 $i+1$ 号点的父亲。

接下来 n 行，每行 L 个整数，第 i 行的第 j 个整数表示 $w[i][j]$ 。

接下来 q 行，每行第一个整数为要修改权值的点 u ，接下来 L 个整数表示新的 $w[u][i]$ 。

3.3 Output

输出到文件 *decompose.out* 中.

输出 q 行, 每行一个整数表示对应询问的答案.

3.4 Sample1

3.4.1 Input

```
4 2 2
1 2 2
1 0
-2 5
1 100
1 100
1 -5 10
2 -16 0
```

3.4.2 Output

```
10
-3
```

3.4.3 Explanation

对于第一组询问, 一种剖分方法为 $\{2, 1\}, \{3\}, \{4\}$.

对于第二组询问, 一种剖分方法为 $\{1\}, \{4, 2\}, \{3\}$.

3.5 Sample2

见选手目录下的 *decompose/decompose2.in* 与 *decompose/decompose2.ans*.

3.6 Sample3

见选手目录下的 *decompose/decompose3.in* 与 *decompose/decompose3.ans*.

3.7 Subtasks

对于所有数据, 有 $1 \leq n \leq 10^5, 2 \leq L \leq 4, 1 \leq f_i \leq i - 1, |w_{i,j}| \leq 10^9, 1 \leq q \leq 10^5, q \times (L - 1) \leq 10^5, 1 \leq u \leq n$.

- subtask1(13%), $n, q \leq 10$.
- subtask2(7%), $q \leq 10$.
- subtask3(8%), f_i 在 $[1, i - 1]$ 内均匀随机.
- subtask4(26%), $f_i = i - 1, L = 2$.
- subtask5(11%), $f_i = i - 1$.
- subtask6(35%), 没有特殊的约束.