

# IOI2004 中国国家队选拔赛

## 第二试

题目名称	网络改造	零件装配	公式编辑器
目录	day2/network	day2/assembly	day2/math
可执行文件名	network	---	maths
输入文件名	network.in	assembly1.in~assembly10.in	maths.in
输出文件名	network.out	assembly1.out~assembly10.out	maths.out
测试点个数	10	10	10
满分	100	100	100
运行限制	10s	---	1s
	128M	---	64M

**竞赛时间：2004 年 5 月 11 日上午 8:15-13:15**

# 网络改造

(network)

## 【背景描述】

HURRICANE小组原来构建的网络由网络上的交换机及其间的网路构成。交换机分级连接，最高级的为顶级的网关交换机，其他交换机分级相连到该网关交换机上。但值得注意的是，任一台非网关交换机与一台高一级的交换机直接相连。而任一台交换机均可以与几台低一级的交换机直接相连。

但最近，由于原来架设的网络服务有限，需要把网络中的一些交换机（包括网关交换机）升级为核心交换机。由于改造的时间所限，只来得及把不超过 $p$ 台（含 $p$ 台）交换机升级为核心交换机，而所有剩下的交换机则需要通过改造网路的方法和这几台核心交换机直接连接。

但是无论是升级交换机还是改造网络都需要花费一定的资金。现在请你给出一个改造网络的方案。使得按照该方案升级后每一个交换机要么是核心交换机，要么直接和核心交换机相连。并且要求提供的方案使改造所用的总费用最小。

## 【任务描述】

你的程序必须根据给定的输入，给出符合题意的输出：

- 输入包括网络的拓扑结构，升级网络中每台交换机的费用，以及改造网络的费用，还有可以升级的交换机的最大数目 $p$ ；
- 你必须根据输入，找出一个升级的方案，满足升级后的核心交换机的数目不超过给定的可升级交换机最大值 $p$ ，且使得总费用最少；
- 其中总费用的计算包括两个部分：
  - 一部分是升级交换机为核心交换机所需要的费用，该部分的费用按照所有的需要升级的交换机所需的费用之和来计算；
  - 另一部分是改造网络所需要的费用，该部分的费用按照所有未升级的交换机到最近的核心交换机的网络路径距离之和来计算；
  - **注意：**当网络中没有任何交换机升级到核心交换机的时候，由于也没有交换机可以连接到核心交换机，所以我们定义此时的总费用为无穷大。

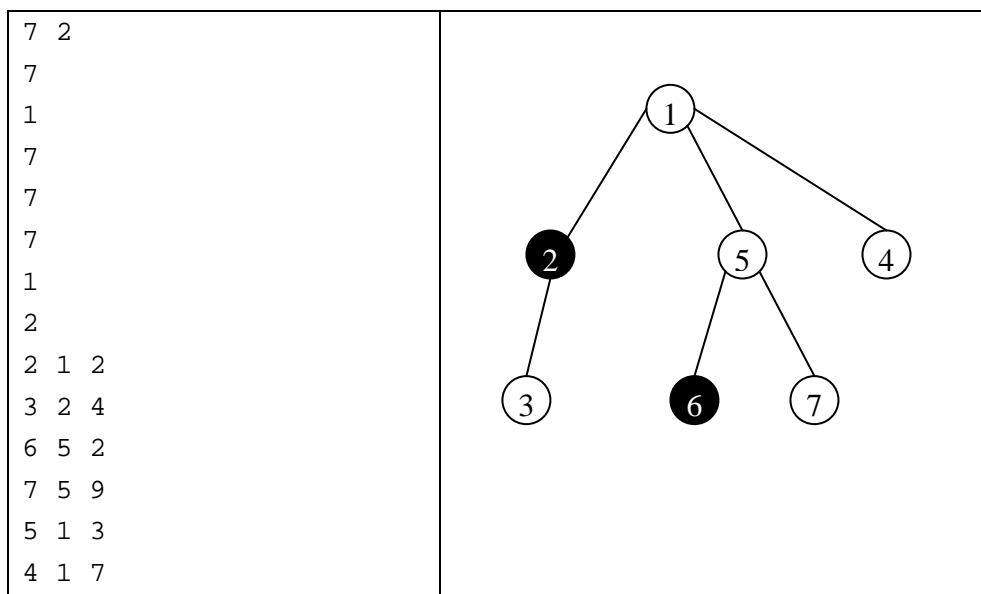
## 【输入格式】：(network.in)

第一行为两个正整数 $n$  ( $n \leq 400$ ) 和 $p$ ，分别表示网络中交换机的数目（交换机按照 1 到 $n$ 标号）和可升级交换机的最大值。接下来的 $n$ 行每行一个正整数 $c_i$ ，表示把标号为 $i$ 的交换机升级为核心交换机所需要的费用。

接下来的 $n-1$ 行每行三个正整数 $i$ 、 $j$ 、 $d_{ij}$  ( $d_{ij} < 20000$ )，表示编号为 $j$ 的交换

机为编号为 $i$ 的交换机的上层交换机，而 $d_{ij}$ 表示两台交换机之间的网路距离。

【输入样例】



【输出格式】: (network.out)

你的输出第一行为一个整数 $M$ ，表示你的方案的最小总费用。接下来一行包括一个整数 $p_0$ ，表示你的方案所需要升级为核心交换机的交换机数目。

【输出样例】

30 2
---------

【运行限制】

运行时间	10 秒
内存使用	128M

【评分方法】

本题目一共有十个测试点，每个测试点的分数为总分数的 10%。对于每个测试点来说，如果你的答案正确，那么你将得到该测试点全部的分数，否则得 0 分。

注意，本题目的测试数据中有 8 个数据的  $n$  不超过 180。

# 零件装配（提交答案）

(assembly)

## 【背景描述】

话说HURRICANE小组在实习中切割好了全部所需的  $n \times m$  个不同的零件之后，又需要把他们装配在一起成为可以使用的产品。现已知每个产品均由  $m$  个部分组成，每个部分恰好为一个零件。由于结构上的差异，在装配产品的某一个部分时，只能选为该部分设计出来的一个零件（从  $n$  个零件中任选一个进行安装，注意：每个零件的安装时间可能不同；且每个零件只能安装一次），不能选为其他部分所设计的零件。

工厂内有  $n$  条装配线可以同时开始  $n$  个产品的装配过程。考虑到不同的零件可能需要不同的装配时间，可以选择适当的装配方案，使得最晚完成产品的装配线所用时间尽可能少。

现在，你的任务就是找到一个这样的装配方案。

## 【任务描述】

你的程序必须根据我们给出的 10 个输入，提供符合题意的 10 个输出：

- 输入中包括需要完成的产品数目  $n$ 、组成产品的部分数  $m$  以及每个部分中的各个零件的安装时间；
- 你需要找出一个方案使得每个零件恰好作为一个部分装配到某一个产品中，且所有的产品中最长的装配时间最短。我们定义一个产品的装配时间为它所有  $m$  部分的零件的装配时间之和；
- 然后你需要把该方案按照下面所约定的格式输出到对应的输出文件中。

## 【输入格式】：(assembly1.in~assembly10.in)

我们提供的输入文件为 assembly1.in~assembly10.in。每个文件的第一行为两个整数，分别代表需要完成产品的数目  $n$  和产品的部分数  $m$ 。

接下来  $n$  行每行  $m$  个非负整数组成一个  $n$  行  $m$  列的矩阵。该矩阵第  $j$  列的数表示第  $j$  个部分的  $n$  个零件每个所需的装配时间。

### 【输入样例】

3 3	该输入表示要完成三个产品，每个产品包括三个部分。
5 4 3	第一部分使用的三个零件所需装配用时分别为 5、3、4。
3 0 5	第二部分使用的三个零件所需装配用时分别为 4、0、3。
4 3 0	第三部分使用的三个零件所需装配用时分别为 3、5、0。

**【输出格式】: (*assembly1.out~assembly10.out*)**

你需要对于我们的输入文件分别完成 *assembly1.out~assembly10.out*。每个文件的第一行只包括一个整数，表示最后完成任务的装配线用时。

接下来  $n$  行每行  $m$  个非负整数为一个  $n$  行  $m$  列的矩阵，第  $i$  行第  $j$  列的数表示第  $i$  条装配线装配第  $j$  个部分时所用零件的装配时间。

**【输出样例】**

9	最迟完成的装配线的用时。
5 4 0	第一条装配线需要装配三个零件，用时分别为 5、4、0。
4 0 5	第二条装配线需要装配三个零件，用时分别为 4, 0, 5。
3 3 3	第三条装配线需要装配三个零件，用时分别为 3, 3, 3。

**【评分方法】**

这是一道提交答案的题目，对于提供的 10 个输入文件，你需要提交相应的输出文件。我们将按照上述的格式对你提交的输出进行检查，如果不符合格式，那么得 0 分，否则我们按照如下的标准对你提交的输出进行评分：

- 如果你的输出方案与输出的装配时间不符，那么得 0 分；
- 我们对于每个测试点规定了一个最长装配时间，如果你程序的输出的装配时间比这个时间要多，那么得 0 分；
- 否则，你的程序将按照下述公式进行评分：

$$score = \left\lfloor \frac{zero\_limit - your\_result}{zero\_limit - best\_result} \times 10\% \right\rfloor$$

其中 *zero\_limit* 为我们规定的最长装配时间，*your\_result* 为你的装配时间，*best\_reuslt* 为已知的最好装配时间。

# 公式编辑器

(maths)

## 【背景描述】

编辑数学公式总是一件烦人的事情，为此 HURRICANE 小组准备出一个编辑数学公式的软件。除了基本功能外，软件还将实现分式和矩阵输入。按照设想，软件应该是符合人性化设计的，必须最大限度的方便用户输入，尽管代价是软件开发极其复杂。幸好输入的方式和格式都已定好，你只需要编个处理程序就行了，下面是相关的约定和格式：

## 【格式控制的概念】

元素	元素可以是 <u>运算符</u> 、 <u>括号</u> 、 <u>数字</u> 、 <u>字母</u> 、 <u>矩阵</u> 、 <u>分式</u> 。
数字、字母和括号	数字包括'0'..'9'、'.'。字母包括'A'..'Z'，'a'..'z'。括号包括'(',')'。它们都只占一行，该行也是对齐行。
表达式	由 0 个到至多 500 个元素构成的序列。其中矩阵与分式元素总个数最多不超过 30 个。
编辑框	一个输入 <u>表达式</u> 的矩形区域。每一个编辑框都包括一个对齐行用作编辑框相互之间的对齐。且我们定义编辑框的宽度为最长一行的字符个数，例如表达式空的时候为 0，高度为最高行与最低行之间相差的行数（包括），但最小为 1，即使表达式为空。
对齐行	<u>编辑框</u> 或 <u>元素</u> 中某一 <u>特定行</u> 。该行用于框内对齐 <u>表达式</u> 及框间对齐，对齐时需要使 <u>表达式</u> 中的 <u>元素</u> 的 <u>对齐行</u> 位于 <u>编辑框</u> 的对齐行上。
运算符	有'+', '-', '*', '/'四个，为了区分“-”（减号）与分数线，“-”两边分别加上一个空列。
矩阵	一个 $m \times n$ ( $1 \leq m \leq 10$ , $1 \leq n \leq 10$ ) 的矩阵包含 $m \times n$ 个编辑框将矩阵分为了 $m$ 个 <u>编辑列</u> 和 $n$ 个 <u>编辑行</u> 。在矩阵的同一 <u>编辑行</u> 中，相邻的两个编辑框按编辑框的对齐行对齐，同一编辑列的编辑框按照它们的宽度居中对齐。且必须保证行与行之间至少存在一个空行，列与列之间至少存在一个空列。第一列左边与最后一列右边每行各有一个“[”和“]”，位于各行表达式的对齐行上，如图所示：

	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>图形表示</p> <math display="block">\left[ \begin{array}{r} 456 \qquad \frac{612}{123} \\ \frac{65535}{1} \qquad 234 \\ 255 \end{array} \right]</math> </div> <div style="text-align: center;"> <p>文本表示</p> <pre>       612       --- [ 456 ]       123       --- 65535 [-----] 234       1       ---       255           </pre> <p>-第一编辑行的对齐行 -空行、整个矩阵的对齐行 -第二编辑行的对齐行</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">间隔 1 列</div> </div> </div> <p>若矩阵的编辑行数 <math>n</math> 为奇数，则矩阵的对齐行为中间那行编辑行中编辑框的对齐行，否则为中间两个编辑行之间的空行；</p>
<b>分式</b>	<p>分式由分子和分母两个编辑框以及它们之间的分数线组成。分式分数线为一条由“-”组成的字符序列，同时它也是分式的对齐行。分式的宽度为两个编辑框宽度的最大值加 2，即在左右两边分别加上一个“-”；而高度为两个编辑框的高度加上分数线的高度 1。两个编辑框按居中对齐，如图：</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>XXXX</span> <span>XX</span> </div> <div style="border-top: 1px solid black; height: 10px; margin: 5px 0;"></div> <div style="display: flex; justify-content: space-between; width: 100%;"> <span>XXX</span> <span>XXXXXX</span> </div> </div> <p>在居中对齐时，如果不能正好对准，则往左偏半格，如上图左边的分母编辑框。</p>

### 【光标控制】

应该指出，编辑框是可以层层嵌套的，比如一个编辑框内有一个矩阵，矩阵内又有若干个编辑框。我们说该编辑框比矩阵的编辑框高一级，矩阵内的所有编辑框同级，分式的两个编辑框也是同级的。**注意：同级只是对一个矩阵或一个分式内的编辑框而言。**

光标可以跳到编辑框的开始和末尾，也可以向四个方向移动，设光标所处的最低一级的编辑框为 A。

- 如果光标跳到编辑框的开始（末尾），则把光标置于 A 的前端（末端）；

➤ 当光标上下移动时

- 如果 A 上(下)方有与 A 同级的编辑框 B, 则把光标置于 B 的前端,

示例	下移一次后
XXXX  ----- XX	XXXX -----  XX

- 否则对比 A 高一级的编辑框作同样判断, 若 A 是最高级别的编辑框, 则不作任何处理。如图, 竖线代表光标:

示例	下移一次后	再下移一次后 (不变)
d ---  c ----- a --- b	d --- c -----  a --- b	d --- c -----  a --- b

➤ 当光标左(右)移动时

- 若光标位于 A 的前端(末端),

- ◆ 如果左(右)边没有同级的编辑框, 则光标将返回到更高一级的编辑框 (若无则不作处理), 置于矩阵或分式的左(右)边;
- ◆ 如果左(右)边有同级的编辑框 B, 则置于 B 的末端 (前端);

示例	左移一次后	再左移两次后
1+[B  A]	1+[B] A]	1+ [B A]

- 若光标左(右)边是一个矩阵或分式,

- ◆ 如果是分式, 则光标置于分子编辑框的末端(前端),

示例	左移一次后
XXXX  ----- XX	XXXX  ----- XX

- ◆ 如果是矩阵, 则置于第  $m$  列  $\left[\frac{n}{2}\right]$  行 (第 1 列  $\left[\frac{n}{2}\right]$  行) 编辑框的末端(前端);

示例	右移一次后
----	-------



	612		612
[	456	-----]	[ 456 -----]
	123		123
	65535		65535
[-----	234	]	[----- 234 ]
	1		1
-----			-----
	255		255

- 若光标左（右）边是其它元素，则光标左（右）移一格。

### 【输入控制】

程序通过一系列的事件来实现输入，每个事件已经被转化成一个字符串：

- ◆ 若字符串只含一个字符，则必为运算符、括号、数字或字母。此时程序在光标处插入该字符，然后光标右移一次；
- ◆ 若字符串为 **Matrix** 或 **Fraction**，则在光标处插入一个1×1的矩阵或分式，光标右移一次；
- ◆ 若字符串为 **AddRow**或**AddCol**，则在矩阵内光标所处的编辑框前插入一行编辑行或一列编辑列，并把光标置于新插入的行（列）中，若光标不在任何矩阵内，则不作处理；
- ◆ 字符串 **Home**、**End**、**Left**、**Right**、**Up**、**Down** 则分别代表光标置于编辑框的开始、末尾，光标向左、右、上、下移动一次。

**注意：**由于该软件只用于输入表达式，并不对表达式进行任何处理，所以**所输入的表达式可能并不正确**。

### 【任务描述】

你需要根据给定的输入，给出满足题意的输出：

- 输入中包括用如上所述格式所描述的操作序列。序列中可能包括运算符、括号、数字、字母、矩阵或分数的插入及光标的移动控制；
- 你需要根据输入的操作序列，完成对表达式的输入；我们约定初始时的表达式为空；
- 然后你需要在输出文件中用我们约定的格式打印出这个表达式。注意表达式行末不要有空格。

## 【输入格式】: (maths.in)

每行有一个代表事件的字符串，直到文件结束。我们约定矩阵的规模不超过  $10 \times 10$ ，总的元素个数不超过 10,000。

### 【输入样例 1】

-	AddCol	Matrix	$-5\frac{1}{6} * \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$
5	AddCol	AddRow	
Fraction	1	AddRow	
1	Right	1	
Down	2	Down	
6	Right	2	
Right	3	Down	
*	Right	3	
Matrix	*		

### 【输入样例 2】

1	Down	Right	$1 + \frac{1+2-2}{1 + \frac{1}{1 + \frac{1}{x}}}$
+	1	Right	
Fraction	+	Home	
1	Fraction	Up	
Down	1	End	
1	Down	+	
+	x	2	
Fraction	Up	-	
1	Up	2	

## 【输出格式】: (maths.out)

按规定的格式输出编辑框的结果，空白的地方用空格补上，每行行末不能有多余空格。最后一行行末保留一个回车。

### 【输出样例 1】

	[1]
1	
- 5---	*[1 2 3]*[2]
6	
	[3]

【输出样例 2】

```
1+2 - 2
1+-----
      1
1+-----
      1
1+----
      x
```

【运行限制】

运行时间	1 秒
内存使用	64M

【评分方法】

本题目一共有十个测试点，每个测试点的分数为总分数的 10%。对于每个测试点来说，如果你的答案完全符合标准答案（注意空格和回车），那么你将得到该测试点全部的分数，否则得 0 分。

**提示：** 在所有的数据中，有三个数据不包括分式，有三个数据不包括矩阵。

【样例说明】

【样例说明 1】

-	-	2	$\frac{1}{-5---*[1\ 2\ ]}$
5	- 5	Right	$\frac{1}{-5---*[1\ 2\ ]}$
Fraction	- 5--	3	$\frac{1}{-5---*[1\ 2\ 3]}$
1	- 5---  1	Right	$\frac{1}{-5---*[1\ 2\ 3]}\Big $
Down	- 5---  1	*	$\frac{1}{-5---*[1\ 2\ 3]*}\Big $

6	$\begin{array}{r} 1 \\ - 5 \text{---} \\ 6 \end{array} \bigg $	Matrix	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [ ] \\ 6 \end{array}$
Right	$\begin{array}{r} 1 \\ - 5 \text{---} \\ 6 \end{array} \bigg  \quad \begin{array}{l} [ ] \\ [ ] \end{array}$	AddRow	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [ ] \\ 6 \end{array} \quad \begin{array}{l} [ ] \\ [ ] \end{array}$
*	$\begin{array}{r} 1 \\ - 5 \text{---} * \\ 6 \end{array} \bigg  \quad \begin{array}{l} [ ] \\ [ ] \end{array}$	AddRow	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [ ] \\ 6 \end{array} \quad \begin{array}{l} [ ] \\ [ ] \end{array}$
Matrix	$\begin{array}{r} 1 \\ - 5 \text{---} * [ ] \\ 6 \end{array}$	1	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [ ] \\ 6 \end{array} \quad \begin{array}{l} [1] \\ [ ] \\ [ ] \end{array}$
AddCol	$\begin{array}{r} 1 \\ - 5 \text{---} * [ ] \\ 6 \end{array}$	Down	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [ ] \\ 6 \end{array} \quad \begin{array}{l} [1] \\ [ ] \\ [ ] \end{array}$
AddCol	$\begin{array}{r} 1 \\ - 5 \text{---} * [ ] \\ 6 \end{array}$	2	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [2] \\ 6 \end{array} \quad \begin{array}{l} [1] \\ [ ] \\ [ ] \end{array}$
1	$\begin{array}{r} 1 \\ - 5 \text{---} * [1] \\ 6 \end{array} \quad \begin{array}{l} [ ] \\ [ ] \end{array}$	Down	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [2] \\ 6 \end{array} \quad \begin{array}{l} [1] \\ [ ] \\ [ ] \end{array}$
Right	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ ] \\ 6 \end{array}$	3	$\begin{array}{r} 1 \\ - 5 \text{---} * [1 \ 2 \ 3] * [2] \\ 6 \end{array} \quad \begin{array}{l} [1] \\ [ ] \\ [3] \end{array}$

## 【样例说明 2】

1	$1 \bigg $	Down	$\begin{array}{r} 1 \\ 1+ \text{---} \text{---} \text{---} \\ 1 \\ 1+ \text{---} \text{---} \text{---} \\ 1 \\ 1+ \text{---} \text{---} \text{---} \\   \end{array}$
+	$1+ \bigg $	x	$\begin{array}{r} 1 \\ 1+ \text{---} \text{---} \text{---} \\ 1 \\ 1+ \text{---} \text{---} \text{---} \\ 1 \\ 1+ \text{---} \text{---} \text{---} \\ x \bigg  \end{array}$

Fraction	$\begin{array}{r}   \\ 1+--- \end{array}$	Up	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\  1 \\ 1+--- \\ x \end{array}$
1	$\begin{array}{r} 1  \\ 1+--- \end{array}$	Up	$\begin{array}{r} 1 \\ 1+----- \\  1 \\ 1+----- \\ 1 \\ 1+--- \\ x \end{array}$
Down	$\begin{array}{r} 1 \\ 1+--- \\   \end{array}$	Right	$\begin{array}{r} 1 \\ 1+----- \\ 1  \\ 1+----- \\ 1 \\ 1+--- \\ x \end{array}$
1	$\begin{array}{r} 1 \\ 1+--- \\ 1  \end{array}$	Right	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+--- \\ x \end{array} \quad \left  \right.$
+	$\begin{array}{r} 1 \\ 1+--- \\ 1+  \end{array}$	Home	$\begin{array}{r} 1 \\ 1+----- \\  1+----- \\ 1 \\ 1+--- \\ x \end{array}$
Fraction	$\begin{array}{r} 1 \\ 1+----- \\   \\ 1+--- \end{array}$	Up	$\begin{array}{r}  1 \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+--- \\ x \end{array}$
1	$\begin{array}{r} 1 \\ 1+----- \\ 1  \\ 1+--- \end{array}$	End	$\begin{array}{r} 1  \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+--- \\ x \end{array}$

Down	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\   \end{array}$	+	$\begin{array}{r} 1+  \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+----- \\ x \end{array}$
1	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\ 1  \end{array}$	2	$\begin{array}{r} 1+2  \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+----- \\ x \end{array}$
+	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\ 1+  \end{array}$	-	$\begin{array}{r} 1+2 -   \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+----- \\ x \end{array}$
Fraction	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\   \\ 1+----- \end{array}$	2	$\begin{array}{r} 1+2 - 2  \\ 1+----- \\ 1 \\ 1+----- \\ 1 \\ 1+----- \\ x \end{array}$
1	$\begin{array}{r} 1 \\ 1+----- \\ 1 \\ 1+----- \\ 1  \\ 1+----- \end{array}$		