

新冠疫情物资分配

2020年初新冠性病毒席卷全国，全国各城市对物资需求突然上涨，特别是口罩（surgical mask=SM）和消毒液（disinfectant=Dis），还有其它（other）一些生活物资。某救助机构接到上级指示救助四个城市的任务。但购买物资资金池（pool）有限，不能完全满足所有城市的需求。救助机构特委托诺基亚编写一套算法，来计算分给每个城市分配资金。最基本要求：按需分配，把可用资金用完。

诺基亚算法部门很快设计出了一套物资分配算法，需要你来实现这个算法，代码要求正确和简洁（clean code）

1. 算法流程：

1. 救助机构设置和每个城市上报([输入文本说明](#))
 1. 救助机构管理员设置物资资金池中总资金和资金池中最小资金的分配比例
 2. 每个城市上报最大需求以及一周内各种物资的实际消耗资金
2. 分配算法：
 1. 给每个城市[分配最小资金](#)
 2. 给每个城市[分配口罩共享资金](#)
 3. 给每个城市[分配消毒液共享资金](#)
 4. 给每个城市[分配其它物资共享资金](#)
 5. 给每个城市算出[分配给每个城市的总资金](#)
 6. 为了精度统一，把总资金做取整处理，参见：[总资金取整算法](#)
3. 输出每个城市分配到的总资金 ([输出文本说明](#))

2. 分配算法

2.1 分配最小资金

首先，为了每个城市都能得到救助，要先保证每个城市都能得到一个最小的资金，某个城市*i*某次*t*分配最小资金 $R_{min}(i, t)$ 的算法如下：

$$R_{min}(i, t) = \min \left\{ \left[R_{pool} \times \frac{PoolMinRate}{100} \times \frac{R_{max}(i, t)}{\sum_{j=0}^{N_{pool}-1} R_{max}(j, t)} \right], R_{max}(i, t) \right\}$$

2.1.1 公式参数说明：

i: 代表城市，索引范围（0, 1, 2, 3）

t: 代表是相对*t*-1而言的，*t*代表本次计算，*t*-1代表是上一次的计算

R_{pool} : 资金池中总资金

poolMinRate: 资金池中用于分配最小资金的比例，比如*poolMinRate*=40 代表有40%的资金用于分配最小资金。范围：[1, 99]

$R_{max}(i, t)$: 第*i*个城市需要的最大资金，最终分配不得超过这个最大资金

N_{pool} : pool中资金服务的城市数量

$\sum_{j=0}^{N_{pool}-1} R_{max}(j, t)$:所有城市上报的最大资金需求，每周每个城市上报

2.1.2 公式文字说明：

最小资金算法就是把pool的资金乘以最小资金百分比，再乘以每个城市最大资金占所有城市的百分比，向上取整，最后和城市最大资金取min，防止超过城市的最大资金。

2.1.3 举例说明：

以两个城市为例，后面的公式也都用**两个城市**为例

参数	值
$poolMinRate$	40
R_{pool}	100
$R_{max}(0, t)$	60
$R_{max}(1, t)$	70

$$R_{min}(0, t) = \min(\text{roundup}(100 * 40 / 100 * 60 / (60 + 70), 60) = 19$$

$$R_{min}(1, t) = \min(\text{roundup}(100 * 40 / 100 * 70 / (60 + 70), 70) = 22$$

2.2 分配口罩共享资金

2.2.1 口罩超额因子

为了让每个城市口罩需求量大时，有机会分到更多的资金；也为了让每个城市因为口罩需求量小时，都可以把资金让给其它城市。所以定义了某个城市*i*某次*t*分配口罩超额因子 $O_{SM}(i, t)$ ：

$$O_{SM}(i, t) = \begin{cases} 1, & \text{if } Load_{SM}(i, t-1) \times O_{SM}(i, t-1) < R_{sum}(i, t-1) \\ overbookFactorSM, & \text{if } Load_{SM}(i, t-1) \times O_{SM}(i, t-1) \geq R_{sum}(i, t-1) \end{cases}$$

2.2.1.1 公式参数说明：

$overbookFactorSM$: 口罩的超额因子，配置固定1.5

$Load_{SM}(i, t-1)$: 上一周内第*i*个城市口罩资金消耗，每次计算前由其城市上报

$O_{SM}(i, t-1)$: 上一计算时的超额因子，初始值为1

$R_{sum}(i, t-1)$: 上一次计算分给第*i*个城市的总资金，初始值为0

2.2.1.2 公式文字说明：

如果上一周城市上报的口罩资金乘以上次计算的超额因子比上一次给这个城市分配的总资金还大，说明这个城市大量的需要口罩，就多给其分配一些资金，这时就给这个城市口罩的超额因子就等于 $overbookFactorSM$ （大于1的值）。否则口罩超额因子的值就是1，公平分配。

2.2.1.3 举例说明：

参数	值
$overbookFactorSM$	1.5
$Load_{SM}(0, t - 1)$	35
$O_{SM}(0, t - 1)$	1.5
$R_{sum}(0, t - 1)$	50
$Load_{SM}(1, t - 1)$	20
$O_{SM}(1, t - 1)$	1
$R_{sum}(1, t - 1)$	50

$$Load_{SM}(0, t - 1) \times O_{SM}(0, t - 1) = 35 \times 1.5 = 52.5$$

$$R_{sum}(0, t - 1) = 50$$

$$Load_{SM}(0, t - 1) \times O_{SM}(0, t - 1) \geq R_{sum}(0, t - 1) \text{ 成立,}$$

$$\text{所以 } O_{SM}(0, t) = overbookFactorSM = 1.5$$

$$Load_{SM}(1, t - 1) \times O_{SM}(1, t - 1) = 20 \times 1 = 20$$

$$R_{sum}(1, t - 1) = 50$$

$$Load_{SM}(1, t - 1) \times O_{SM}(1, t - 1) < R_{sum}(1, t - 1) \text{ 成立,}$$

$$\text{所以 } O_{SM}(1, t) = overbookFactorSM = 1$$

2.2.2 口罩专有资金

因为每个城市已经分配了最小资金，是肯定要分给每个城市的，所以这部分资金称为**专有资金**，某个城市*i*某次*t*分配专有资金 $R_{dedicated,SM}(i, t)$ 算法如下：

$$R_{dedicated,SM}(i, t) = \min \{O_{SM}(i, t) \times Load_{SM}(i, t - 1), R_{min}(i, t)\}$$

2.2.2.1 公式参数说明：

$O_{SM}(i, t)$: 本次计算口罩资金使用的超额因子

$Load_{SM}(i, t - 1)$: 上一周内第*i*个城市口罩资金消耗，每次计算前由其城市上报

$R_{min}(i, t)$: 第*i*个城市的最小资金

2.2.2.2 公式文字说明：

口罩专有资金就等于上一周期内每个城市上报的口罩消耗资金乘以超额因子，这个乘积代表着这个城市的需求，由于算的是专有资金，所以不能超过最小资金

2.2.2.3 举例说明：

参数	值
$R_{min}(0, t)$	19
$O_{SM}(0, t)$	1.5
$Load_{SM}(0, t - 1)$	35
$R_{min}(1, t)$	22
$O_{SM}(1, t)$	1
$Load_{SM}(1, t - 1)$	20

$$R_{dedicated, SM}(0, t) = \min(1.5 * 35, 19) = 19$$

$$R_{dedicated, SM}(1, t) = \min(1 * 20, 22) = 20$$

2.2.3 口罩理想共享资金

共享资金的意思是除了已经分配的最小资金，pool中还剩下的的资金，要被每个城市共享。

理想的意思是指是不考虑其它城市，只考虑自己城市的需求

某个城市*i*某次*t*分配口罩理想共享资金 $R_{idealShardSM}(i, t)$ 算法如下：

$$R_{idealShardSM}(i, t) = \max \{ \min \{ O_{SM}(i, t) \times Load_{SM}(i, t - 1) - R_{dedicated, SM}(i, t), (R_{max}(i, t) - R_{min}(i, t)) \}, 0 \}$$

2.2.3.1 公式参数说明：

$O_{SM}(i, t)$: 本次计算口罩资金使用的超额因子

$Load_{SM}(i, t - 1)$: 上一周内第*i*个城市口罩资金消耗，每次计算前由其城市上报

$R_{dedicated, SM}(i, t)$: 口罩专有资金

$R_{min}(i, t)$: 第*i*个城市的最小资金

$R_{max}(i, t)$: 第*i*个城市需要的最大资金，最终分配不得超过这个最大资金

2.2.3.2 公式文字说明：

第*i*个城市的口罩理想共享资金就等于，这个城市的需求（上个周期口罩的消耗 × 超额因子） - 专有资金（从最小资金中得到）

但这个城市已经分配了最小资金，所以再分pool中还剩下的资金时，要考虑总得到的资金不能超过这个城市所需的最大资金。

因为有减法，保护一下理想共享资金不能为负值

2.2.3.3 举例说明：

参数	值
$R_{min}(0, t)$	19
$O_{SM}(0, t)$	1.5
$Load_{SM}(0, t - 1)$	35
$R_{dedicated, SM}(0, t)$	19
$R_{max}(0, t)$	60
$R_{min}(1, t)$	22
$O_{SM}(1, t)$	1
$Load_{SM}(1, t - 1)$	20
$R_{dedicated, SM}(1, t)$	20
$R_{max}(1, t)$	70

$$R_{idealShardSM}(0, t) = \max\{\min((1.5 * 35 - 19), (60 - 19)), 0\} = 31$$

$$R_{idealShardSM}(1, t) = \max\{\min((1 * 20 - 20), (70 - 22)), 0\} = 0$$

2.2.4 口罩共享资金

理想共享资金只考虑自己，但pool中的总资金是有限的，所以要按每个城市所需的理想共享资金的比例来分配pool中剩余的资金，某个城市*i*某次*t*分配口罩共享资金 $R_{sharedSM}(i, t)$ 算法如下：

$$R_{sharedSM}(i, t) = \min \left\{ R_{idealSharedSM}(i, t), \frac{R_{idealSharedSM}(i, t)}{\sum_{j=0}^{N_{pool}-1} R_{idealSharedSM}(j, t)} \times R_{SM, pool}(t) \right\}$$

2.2.4.1 公式参数说明：

$R_{idealShardSM}(i, t)$: 口罩理想共享资金

N_{pool} : pool中资金服务的城市数量

$\sum_{j=0}^{N_{pool}-1} R_{idealSharedSM}(j, t)$: 所有城市的口罩理想共享资金之和

R_{pool} : 资金池中可用的资金

$R_{SM, pool}(t) = R_{pool} - \sum_{j=0}^{N_{pool}-1} R_{min}(j, t)$: 资金池中总资金 - 所有城市的最小资金之和，即资金池中

还剩下的可用于分配口罩的资金

2.2.4.2 公式文字说明：

这一步就是每个城市按自己的理想共享资金在所有城市中所占的比例，去瓜分资金池中还剩下的资金（除去每个城市的最小资金），当然真正共享资金不能超过自己的理想共享资金。

2.2.4.3 举例说明：

参数	值
R_{pool}	100
$R_{min}(0, t)$	19
$R_{idealShardSM}(0, t)$	33.5
$R_{min}(1, t)$	22
$R_{idealShardSM}(1, t)$	0

$R_{SM,pool}(t) = 100 - (19 + 22) = 59$

$R_{sharedSM}(0, t) = \min(33.5, 33.5 / (33.5 + 0) * 59) = 33.5$

$R_{sharedSM}(1, t) = \min(0, 0 / (33.5 + 0) * 59) = 0$

2.3 分配消毒液共享资金

2.3.1 消毒液超额因子:

为了让每个城市消毒液需求量大时，有机会分到更多的资金；也为了让每个城市因为消毒液需求量小时，都可以把资金让给其它城市。所以定义了某个城市*i*某次*t*分配消毒液超额因子 $O_{Dis}(i, t)$:

$$O_{Dis}(i, t) = \begin{cases} 1, & \text{if } Load_{Dis}(i, t - 1) \times O_{Dis}(i, t - 1) < R_{sum}(i, t - 1) \\ overbookFactorDis, & \text{if } Load_{Dis}(i, t - 1) \times O_{Dis}(i, t - 1) \geq R_{sum}(i, t - 1) \end{cases}$$

2.3.1.1 公式参数说明:

overbookFactorDis: 消毒液的超额因子，默认配置1.2

$Load_{Dis}(i, t - 1)$: 上一周内第*i*个城市消毒液资金消耗，每次计算前由其城市上报

$O_{Dis}(i, t - 1)$: 上一次计算时的超额因子，初始值为1

$R_{sum}(i, t - 1)$: 上一次计算分给第*i*个城市的总资金，初始值为0

2.3.1.2 公式文字说明:

如果上一周期城市上报的消毒液资金乘以上次计算的超额因子比上一次给这个城市分配的总资金还大，说明这个城市大量的需要消毒液，就多给其分配一些资金，这时就给这个城市消毒液的超额因子就等于*overbookFactorDis*（大于1的值）。否则消毒液超额因子的值就是1，公平分配。

2.3.1.3 举例说明:

参数	值
<i>overbookFactorDis</i>	1.2
$Load_{Dis}(0, t - 1)$	1
$O_{Dis}(0, t - 1)$	1.2
$R_{sum}(0, t - 1)$	50
$Load_{Dis}(1, t - 1)$	17
$O_{Dis}(1, t - 1)$	1.2
$R_{sum}(1, t - 1)$	50

$$Load_{Dis}(0, t - 1) \times O_{Dis}(0, t - 1) = 1 * 1.2 = 1.2$$

$$R_{sum}(0, t - 1) = 50$$

$$Load_{Dis}(0, t - 1) \times O_{Dis}(0, t - 1) < R_{sum}(0, t - 1) \text{ 成立,}$$

$$\text{所以 } O_{Dis}(0, t) = overbookFactorDis = 1$$

$$Load_{Dis}(1, t - 1) \times O_{Dis}(1, t - 1) = 17 * 1.2 = 20.4$$

$$R_{sum}(1, t - 1) = 50$$

$$Load_{Dis}(1, t - 1) \times O_{Dis}(1, t - 1) < R_{sum}(1, t - 1) \text{ 成立,}$$

$$\text{所以 } O_{Dis}(i, t) = overbookFactorDis = 1$$

2.3.2 消毒液专有资金

消毒液的专有资金和口罩的专有资金的公式是一样的，只是经过口罩的分配后，最小资金可能已经被口罩的专有资金占用了一部分，最小资金中还剩下可供消毒液使用的资金 $R_{min,Dis}(i, t)$ 就等于最小资源减去口罩占用的专有资源，某个城市 i 某次 t 分配消毒液专有资金 $R_{dedicated,Dis}(i, t)$ 公式如下：

$$R_{min,Dis}(i, t) = R_{min}(i, t) - R_{dedicated,SM}(i, t)$$

$$R_{dedicated,Dis}(i, t) = \min \{ O_{Dis}(i, t) \times Load_{Dis}(i, t - 1), R_{min,Dis}(i, t) \}$$

2.3.2.1 公式参数说明：

$O_{Dis}(i, t)$: 本次计算消毒液资金使用的超额因子

$Load_{Dis}(i, t - 1)$: 上一周内第 i 个城市消毒液资金消耗，每次计算前由其城市上报

$R_{min,Dis}(i, t)$: 第 i 个城市的最小资金中还剩下可供消毒液的资金

2.3.2.2 公式文字说明：

消毒液专有资金就等于上一周期内每个城市上报的消毒液消耗资金乘以超额因子，这个乘积代表着这个城市的需求，但不能超过最小资金中还剩下可供消毒液的资金

2.3.2.3 举例说明：

参数	值
$R_{min}(0, t)$	19
$O_{Dis}(0, t)$	1
$Load_{Dis}(0, t - 1)$	0
$R_{dedicated,SM}(0, t)$	19
$R_{min}(1, t)$	22
$O_{Dis}(1, t)$	1
$Load_{Dis}(1, t - 1)$	20
$R_{dedicated,SM}(1, t)$	20

$$R_{min,Dis}(0, t) = 19 - 19 = 0$$

$$R_{dedicated,Dis}(0, t) = \min(1 * 0, 0) = 0$$

$$R_{min,Dis}(1, t) = 22 - 20 = 2$$

$$R_{dedicated,Dis}(1, t) = \min(1 * 20, 2) = 2$$

2.3.3 消毒液理想共享资金

消毒液理想共享资金的分配算法和口罩是一样的,只是要考虑这时每个城市已经分配了最小资金和口罩共享资金, 所以每个城市的最大需求也要相应的减去最小资金和口罩共享资金。

某个城市*i*某次*t*分配消毒液理想共享资金 $R_{idealShardDis}(i, t)$ 公式如下:

$$R_{idealShardDis}(i, t) = \max \{ \min \{ O_{Dis}(i, t) \times Load_{Dis}(i, t - 1) - R_{dedicated,Dis}(i, t), (R_{max}(i, t) - R_{min}(i, t) - R_{sharedSM}(i, t)) \}, 0 \}$$

2.3.3.1 公式参数说明:

$O_{Dis}(i, t)$: 本次计算消毒液资金使用的超额因子

$Load_{Dis}(i, t - 1)$: 上一周内第*i*个城市口罩资金消耗, 每次计算前由其城市上报

$R_{dedicated,Dis}(i, t)$: 消毒液专有资金

$R_{sharedSM}(i, t)$: 口罩的共享资金

$R_{min}(i, t)$: 第*i*个城市的最小资金

$R_{max}(i, t)$: 第*i*个城市需要的最大资金, 最终分配不得超过这个最大资金

2.3.3.2 公式文字说明:

第*i*个城市的消毒液理想共享资金就等于, 这个城市的需求 (上周消毒液的消耗 × 超额因子) - 专有资金 (从最小资金中得到)

但这个城市已经分配了最小资金和口罩共享资金, 所以再分pool中还剩下的资金时, 要考虑总得到的资金不能超过这个城市所需的最大资金。

因为有减法, 保护一下理想共享资金不能为负值

2.3.3.3 举例说明:

参数	值
$R_{min}(0, t)$	19
$O_{Dis}(0, t)$	1
$Load_{Dis}(0, t - 1)$	1
$R_{dedicated, Dis}(0, t)$	0
$R_{max}(0, t)$	60
$R_{sharedSM}(0, t)$	33.5
$R_{min}(1, t)$	22
$O_{Dis}(1, t)$	1
$Load_{Dis}(1, t - 1)$	17
$R_{dedicated, Dis}(1, t)$	2
$R_{max}(1, t)$	70
$R_{sharedSM}(1, t)$	0

$$R_{idealShardDis}(0, t) = \max\{\min((1 * 1 - 0), (60 - 19 - 33.5)), 0\} = 1$$

$$R_{idealShardDis}(1, t) = \max\{\min((1 * 17 - 2), (70 - 22 - 0)), 0\} = 15$$

2.3.4 消毒液共享资金

某个城市*i*某次*t*分配消毒液共享资金 $R_{sharedDis}(i, t)$ 公式如下：

$$R_{sharedDis}(i, t) = \min \left\{ R_{idealSharedDis}(i, t); \frac{R_{idealSharedDis}(i, t)}{\sum_{j=0}^{N_{pool}-1} R_{idealSharedDis}(j, t)} \times R_{Dis, pool}(t) \right\}$$

$$R_{Dis, pool}(t) = R_{SM, pool}(t) - \sum_{j=1}^{N_{pool}} R_{sharedSM}(j, t)$$

2.3.4.1 公式参数说明：

$R_{idealShardDis}(i, t)$: 消毒液理想共享资金

N_{pool} : pool中资金服务的城市数量

$\sum_{j=0}^{N_{pool}-1} R_{idealSharedDis}(j, t)$: 所有城市的消毒液理想共享资金之和

$R_{SM, pool}(t)$: 资金池中还剩下的可用于分配口罩的资金

$R_{Dis, pool}(t) = R_{SM, pool}(t) - \sum_{j=0}^{N_{pool}-1} R_{sharedSM}(j, t)$: 资金池中还剩下的可用于分配口罩的资金 - 所有城市的口罩共享资金之和，即资金池中还剩下的可用于消毒液分配的资金

2.3.4.2 公式文字说明：

这一步就是每个城市按自己的理想共享资金在所有城市中所占的比例，去瓜分资金池中还剩下的可用于消毒液分配的资金，当然真正共享资金不能超过自己的理想共享资金。

2.3.4.3 举例说明：

参数	值
$R_{SM,pool}(t)$	59
$R_{sharedSM}(0, t)$	33.5
$R_{idealShardSM}(0, t)$	1
$R_{sharedSM}(1, t)$	0
$R_{idealShardSM}(1, t)$	15

$$R_{Dis,pool}(t) = 59 - (33.5 + 0) = 25.5$$

$$R_{sharedSM}(0, t) = \min(1, 1/(1+15) * 25.5) = 1$$

$$R_{sharedSM}(1, t) = \min(15, 15/(1+15) * 25.5) = 15$$

2.4 分配其它物资共享资金

由于其它物资没有那么重要，所以分配算法相对也简单。共分两步：

1. 其它物资最大共享资金，防止超过每个城市的最大需求
2. 其它物资共享资金，按每个城市其它物资实际使用（考虑其它物资超额因子）的上报比例来分配

2.4.1 其它物资最大共享资金：

某个城市*i*某次*t*分配其它物资最大共享资金 $R_{max,sharedOther}(i, t)$ 公式如下：

$$R_{max,sharedOther}(i, t) = R_{max}(i, t) - R_{min}(i, t) - R_{sharedSM}(i, t) - R_{sharedDis}(i, t)$$

2.4.1.1 公式参数说明：

$R_{max}(i, t)$:第*i*个城市需要的最大资金，最终分配不得超过这个最大资金

$R_{min}(i, t)$: 第*i*个城市的最小资金

$R_{sharedSM}(i, t)$: 第*i*个城市的口罩的共享资金

$R_{sharedDis}(i, t)$: 第*i*个城市的消毒液的共享资金

2.4.1.2 公式文字说明：

其它物资最大共享资金就是这个城市最大需要的最大资金减去已经分配给这个城市的资金（包括最小资金和口罩、消毒液共享资金）

2.4.1.3 举例说明：

参数	值
$R_{max}(0, t)$	60
$R_{min}(0, t)$	19
$R_{sharedSM}(0, t)$	33.5
$R_{sharedDis}(0, t)$	1
$R_{max}(1, t)$	70
$R_{min}(1, t)$	22
$R_{sharedSM}(1, t)$	0
$R_{sharedDis}(1, t)$	15

$$R_{max,sharedOther}(0, t) = 60 - 19 - 33.5 - 1 = 6.5$$

$$R_{max,sharedOther}(1, t) = 70 - 22 - 0 - 15 = 33$$

2.4.2 其他物资超额因子

某个城市*i*某次*t*分配其它物资超额因子 $O_{other}(i, t)$ 公式如下：

$$O_{(other)}(i, t) = \begin{cases} 1, & \text{如果没轮到} \\ overbookFactorOther, & \text{如果轮到} \end{cases}$$

2.4.2.1 公式参数说明：

$overbookFactorOther$: 其它物资的超额因子，默认配置1.5

2.4.2.2 公式文字说明：

其它物资由于不是紧要的资金所以就以轮询的方式，让每个城市都有可能多分一些资金，这样其它物资的需求量多的城市也可以多分一些资金

2.4.2.3 举例说明：

如果上一次计算，城市（0）使用 $overbookFactorOther$ ，即： $O_{other}(0, t - 1) = overbookFactorOther = 1.5$ 其它城市使用1

那么本次计算，城市（1）使用 $overbookFactorOther$ 即： $O_{other}(1, t) = overbookFactorOther = 1.5$ ，其它城市使用1

2.4.3 其它物资共享资金：

某个城市*i*某次*t*分配其它物资共享资金 $R_{sharedOther}(i, t)$ 公式如下：

$$R_{sharedOther}(i, t) = \min \left\{ R_{max,sharedOther}(i, t), \frac{LoadOther(i, t - 1) \times O_{other}(i, t)}{\sum_{j=0}^{N_{pool}-1} (LoadOther(j, t - 1) \times O_{other}(j, t))} \times R_{Other,pool}(i, t) \right\}$$

$$R_{Other,pool}(i, t) = R_{Dis,pool}(i, t) - \sum_{j=0}^{N_{pool}-1} R_{sharedDis}(j, t)$$

2.4.3.1 公式参数说明：

$Load_{Other}(i, t - 1)$: 上一周内第 i 个城市其它物资资金消耗，每次计算前由其城市上报

$O_{other}(i, t)$: 本次计算其它物资资金使用的超额因子，所有城市轮询使用

N_{pool} : pool中资金服务的城市数量

$\sum_{j=0}^{N_{pool}-1} (Load_{other}(j, t) \times O_{other}(j, t))$: 所有城市上报的实际消耗的其它物资乘以其超额因子之和

$R_{Dis,pool}(i, t)$: 资金池中还剩下的可用于分配消毒液的资金

$R_{Other,pool}(i, t) = R_{Dis,pool}(i, t) - \sum_{j=0}^{N_{pool}-1} R_{sharedDis}(j, t)$: 资金池中还剩下的可用于分配消毒液的资

金 - 所有城市的消毒液共享资金之和，即资金池中还剩下的可用于其它物资分配的资金

2.4.3.2 公式文字说明:

这一步就是每个城市其它物资实际使用的上报比例，来瓜分资金池中还剩下的可用于其它物资分配的资金，当然真正其它物资共享资金不能超过其它物资最大共享资金。

2.4.3.3 举例说明:

参数	值
$R_{Dis,pool}(t)$	25.5
$R_{sharedDis}(0, t)$	1
$Load_{Other}(0, t - 1)$	10
$O_{other}(0, t)$	1
$R_{max,sharedOther}(0, t)$	6.5
$R_{sharedDis}(1, t)$	15
$R_{max,sharedOther}(1, t)$	33
$Load_{Other}(1, t - 1)$	10
$O_{other}(1, t)$	1.5

$$R_{other,pool}(t) = 25.5 - (1 + 15) = 9.5$$

$$R_{sharedOther}(0, t) = \min(6.5, (10 * 1) / ((10 * 1) + (10 * 1.5)) * 9.5) = 3.8$$

$$R_{sharedOther}(1, t) = \min(33, (10 * 1.5) / ((10 * 1) + (10 * 1.5)) * 9.5) = 5.7$$

2.4.3.2 公式文字说明:

这一步就是每个城市其它物资实际使用的上报比例，来瓜分资金池中还剩下的可用于其它物资分配的资金，当然真正其它物资共享资金不能超过其它物资最大共享资金。

2.4.3.3 举例说明:

参数	值
$R_{Dis,pool}(t)$	25.5
$R_{sharedDis}(0, t)$	1
$Load_{Other}(0, t - 1)$	10
$O_{other}(0, t)$	1
$R_{max,sharedOther}(0, t)$	6.5
$R_{sharedDis}(1, t)$	15
$R_{max,sharedOther}(1, t)$	33
$Load_{Other}(1, t - 1)$	10
$O_{other}(1, t)$	1.5

$$R_{other,pool}(t) = 25.5 - (1 + 15) = 9.5$$

$$Load_{other,overbook}(0, t - 1) = \max(10 * 1, 0.000001) = 10$$

$$Load_{other,overbook}(1, t - 1) = \max(10 * 1.5, 0.000001) = 15$$

$$R_{sharedOther}(0, t) = \min(6.5, (10 / 25) * 9.5) = 3.8$$

$$R_{sharedOther}(1, t) = \min(33, (15 / 25) * 9.5) = 5.7$$

2.5 分配给每个城市的总资金

2.5.1 计算总资金

分配给城市的总资金 = 最小资金+口罩共享资金+消毒液共享资金+其它物资共享资金

$$R_{sum}(i, t) = R_{min}(i, t) + R_{sharedSM}(i, t) + R_{sharedDis}(i, t) + R_{sharedOther}(i, t)$$

2.5.1.1 举例说明：

参数	值
$R_{max}(0, t)$	60
$R_{min}(0, t)$	19
$R_{sharedSM}(0, t)$	33.5
$R_{sharedDis}(0, t)$	1
$R_{sharedOther}(0, t)$	3.8
$R_{max}(1, t)$	70
$R_{min}(1, t)$	22
$R_{sharedSM}(1, t)$	0
$R_{sharedDis}(1, t)$	15
$R_{sharedOther}(1, t)$	5.7

$$R_{sum}(0, t) = 19 + 33.5 + 1 + 3.8 = 57.3$$

$$R_{sum}(1, t) = 22 + 0 + 15 + 5.7 = 42.7$$

2.6 总资金取整算法

但是，考虑到精度问题，如果是小数那么可能结果会不唯一，所以限制每个城市的总资金只能为整数，如果所有城市都向下取整，资金池中可能还会有剩余；如果都向上取整，资金池中资金不够分。特设计以下算法：

1. 把分个城市分配到的总资金，先取其整数部分。

$$R_{sum}(i, t) = R_{sum, intPart}(i, t) = \lfloor R_{sum}(i, t) \rfloor : \text{总资金的整数部分}$$

2. 如果每个城市只分配整数部分，资金池中可能还有剩余的资金

$$R_{pool, remand}(t) = R_{pool} - \sum_{j=0}^{N_{pool}-1} R_{sum, intPart}(j, t) : \text{如果每个城市只分整数部分，资金池中还剩下的资金}$$

3. 按每个城市的上报的编号（输入中每个城市从前到后分别为0, 1, 2, 3）轮询把剩余的资金分配给城市，但要确保每个城市不能超过最大值 $R_{max}(i, t)$ 。算法伪代码如下：

```
for(k = 0; (k < 4) and (Rpool, remand(t) > 0); k++)
{
    Rsum(k, t) = min { Rsum, intPart(k, t) + Rpool, remand(t), Rmax(k, t) };
    Rpool, remand(t) = Rpool, remand(t) - ( Rsum(k, t) - Rsum, intPart(k, t) )
}
```

2.6.1 举例说明：

参数	值
$R_{sum}(0, t)$	57.3
$R_{sum}(1, t)$	42.7
$R_{pool}(t)$	100
$R_{max}(0, t)$	60
$R_{max}(1, t)$	70

$$R_{sum}(0, t) = R_{sum, intPart}(0, t) = \text{roundDown}(57.3) = 57$$

$$R_{sum}(1, t) = R_{sum, intPart}(1, t) = \text{roundDown}(42.7) = 42$$

$$R_{pool, remand}(t) = 100 - (57 + 42) = 1$$

$$R_{sum}(0, t) = \min(57 + 1, 60) = 58$$

因为： $R_{pool, remand}(t) = 1 - (58 - 57) = 0$ ， 所以不再循环， 城市1只有整数部分， 即： $R_{sum}(1, t) = 42$

3.输入输出

3.1 输入

3.1.1 输入说明：

1. 输入文件位置：在工程目录下的**testcases**目录下，比如case-1-input.txt，输入文件是以命令行参数的形式传入，即第一个参数就是输入文件：

```
$ Executable inputFile
```

文件内容说明如下：

1. 救助机构管理员设置相关参数（输入）

第一行，资金池最大值和分配最小资金的比例，共两个参数，在这个文件中，后面周计算都用此数据，即，每轮计算都有 R_{pool} 资金使用而分配最小资金比例都是 $poolMinRate$ ，数据用","隔开。

$R_{pool}, poolMinRate$

2. 每个城市每周上报最大需求以及各种物资实际消耗情况（输入）

第二行开始，每4行代表4个城市一周的上报，如要是8行就是两周的上报，以此类推第周的上报以空行隔开，每行每个数据用","隔开

$R_{max}(0, t), Load_{SM}(0, t - 1), Load_{Dis}(0, t - 1), Load_{Other}(0, t - 1)$

$R_{max}(1, t), Load_{SM}(1, t - 1), Load_{Dis}(1, t - 1), Load_{Other}(1, t - 1)$

$R_{max}(2, t), Load_{SM}(2, t - 1), Load_{Dis}(2, t - 1), Load_{Other}(2, t - 1)$

$R_{max}(3, t), Load_{SM}(3, t - 1), Load_{Dis}(3, t - 1), Load_{Other}(3, t - 1)$

3.1.2 输入限制

1. 所有城市最大需求之和大于资金池中总资金

3.1.2 举例：

case-1-input.txt

```
100,40
50,40,0,10
60,20,20,10
60,20,0,10
70,0,10,10

50,20,10,10
50,10,10,4
30,6,6,12
40,0,0,10
```

3.2 输出

3.2.1 输出说明：

因为clean code才是此次比赛的考查重点目的，所以正确结果我们是公开的。已经在工程目录下的**testcases**目录中了和输入文件对应，比如case-1-output.txt

但正确性也是要求的，要求选手所编写的程序要在stdout上输出结果(也就是把结果打印到屏幕上)，CI脚本会检查打屏的结果是否和对应的正确结果一致

输出内容说明如下：

每行代表每周救助机构根据每个城市的上报分配给每个城市的总资金，每个值以","隔开

$R_{sum}(0, t), R_{sum}(1, t), R_{sum}(2, t), R_{sum}(3, t)$

3.2.1 举例：

case-1-output.txt

```
40,24,24,12  
37,24,22,17
```

4. clean code要求

