

18_Nachoneko 做衣服 (Nachoneko_and_Beautiful_Clothes)

(1 分/1 分/8 分/15 分)

時間限制: 5 seconds

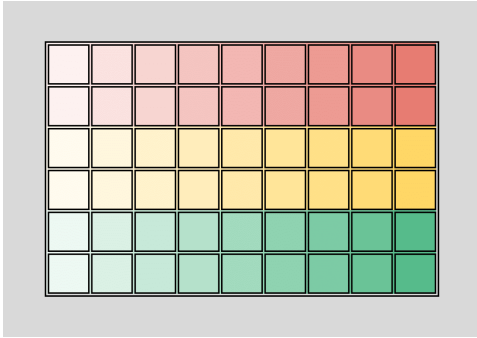
記憶體限制: 1024 MB

題目敘述

Nachoneko 媽媽正在為可愛的孩子們做女裝，而第一步就是要先找出適合的顏色們。

她拿出了一張布料色卡，打算從裡面剪下一些布料來做實驗。

這張布料色卡可以被想像成一個高 h 公分寬 w 公分的矩形，為了方便溝通，我們將其分為 h 個 row 跟 w 個 column，再給由上往下數的第 r 個 row、由左往右數的第 c 個 column 的 1×1 矩形編號為 (r, c) 。



一張 $(h, w) = (6, 9)$ 的布料色卡。

其中最右上角的紅色矩形編號為 $(1, 9)$ 、最左下角的白綠色矩形編號為 $(6, 1)$ 。

現在，Nachoneko 想要從裡面剪下 n 個高度為 1 的矩形，每個矩形可以以三個參數 $(r[i], c[i][0], c[i][1])$ 來表示，代表這個矩形的兩個端點座標是 $(r[i], c[i][0])$ 跟 $(r[i], c[i][1])$ 。愛惜布料的她不希望這 n 個區間以外的布料被剪下來（為此，她只會需要 r 是奇數的那些區間），不過這些區間內的布料可以被剪成任意份。

Nachoneko 給了你三種可以做的操作，你能夠告訴她應該要怎樣才能把布料都剪下來嗎？

1. **combine** $c[0]$ $c[1]$ ：把整張布料的第 $c[0]$ 個 column 到第 $c[1]$ 個 column 壓縮到一起，請注意布料的編號並不會因此而改變。
2. **cut** r c ：把跟編號 (r, c) 的布料重疊的所有布料都剪下來。
3. **spread**：把整張布攤開，你必須且只能執行該操作 1 次，且這必須是你的最後一個操作。

因為 Nachoneko 的體力不好，所以如果你讓她做超過 $q_{\max} = 300\,000$ 次操作她就會躺在地上跟地板融為一體，沒辦法再給你 AC。

輸入格式

- line 1: **subtask**: 代表測資所屬的子任務編號。
- line 2: h w n

- line $3 + i$ ($0 \leq i \leq n - 1$) : $r[i]$ $c[i][0]$ $c[i][1]$

輸出格式

- line 1: q
- line $2 + i$ ($0 \leq i \leq q - 1$): `Operation[i]`
- 你輸出的 q 必須滿足 $1 \leq q \leq q_{\max}$ 。
- `Operation` 的格式以及限制如下：
 1. `combine c[0] c[1]`
 - 你的輸出必須滿足 $1 \leq c[0] \leq c[1] \leq w$ 。
 2. `cut r c`
 - 你的輸出必須滿足 $1 \leq r \leq h$ 及 $1 \leq c \leq w$ 。
 3. `spread`
 - 你必須且只能執行該操作 1 次（即便你沒有進行任何 `combine` 操作），且這必須是你的最後一個操作。
- 請在每一個操作之後換行，且不要在行尾輸出多餘的空格，否則你可能會得到 Wrong Answer。

資料範圍

- $1 \leq \text{subtask} \leq 3$ 。
- $q_{\max} = 300\,000$ 。
- $1 \leq h, w \leq 120\,000$ 。
- $1 \leq n \leq 500$ 。
- $1 \leq r[i] \leq h$ ($0 \leq i \leq n - 1$)。
- $r[i]$ 是奇數 ($0 \leq i \leq n - 1$)。
- $1 \leq c[i][0] \leq c[i][1] \leq w$ ($0 \leq i \leq n - 1$)。
- 輸入的數字都是整數。

子任務

- 子任務 1 滿足 $h \leq 100$ 、 $w \leq 100$ 。
- 子任務 2 滿足 $c[i - 1][1] < c[i][0]$ ($1 \leq i \leq n - 1$)。
- 子任務 3 滿足 $n \leq 100$ 。
- 子任務 4 無額外限制。

在子任務中你可獲得部分分數。令 $q_1, q_2, \dots, q_k$ 是你在該子任務中的每一筆測資所輸出的 q 值； $q'_1, q'_2, \dots, q'_k$ 是該子任務中每一筆測資的理論最小操作次數；且 $Q = \max\{q_1, q_2, \dots, q_k\}$ 。你在四個子任務的得分將依下表計算：

操作次數	子任務 1、2 得分
$6001 \leq Q \leq 300\,000$	0.5
$Q \leq 6000$	1

操作次數	子任務 3 得分
$2001 \leq Q \leq 300\,000$	1
$Q \leq 2000$ 且存在測資使 $q_i > q'_i$	3
$Q \leq 2000$ 且所有測資皆 $q_i \leq q'_i$	8

操作次數	子任務 4 得分
$180\,001 \leq Q \leq 300\,000$	2
$30\,001 \leq Q \leq 180\,000$	4
$10\,001 \leq Q \leq 30\,000$	6
$Q \leq 10\,000$ 且存在測資使 $q_i > q'_i$	10
$Q \leq 10\,000$ 且所有測資皆 $q_i \leq q'_i$	15

測試範例

輸入範例 1

```
1
10 10 5
1 3 7
3 4 5
5 10 10
7 1 10
9 3 9
```

不考慮 `subtask` 的話，該範例輸入符合子任務 1, 3, 4 的限制。

輸出範例 1

```
10
combine 4 5
cut 3 5
combine 3 7
cut 1 5
cut 5 10
combine 6 9
cut 9 8
combine 1 10
cut 7 1
spread
```

可以證明不存在 $q \leq 9$ 的解。

輸入範例 2

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

不考慮 **subtask** 的話，該範例輸入符合子任務 1, 3, 4 的限制。

輸出範例 2

```
3
combine 1 5
cut 1 3
spread
```

可以證明不存在 $q \leq 2$ 的解。

輸入範例 3

```
2
1 100 3
1 1 2
1 3 98
1 99 100
```

輸出範例 3

```
3
combine 1 100
cut 1 1
spread
```

部份分範例

輸入

```
2
1 100 3
1 1 2
1 3 98
1 99 100
```

不考慮 subtask 的話，該範例輸入符合子任務 2, 3, 4 的限制。

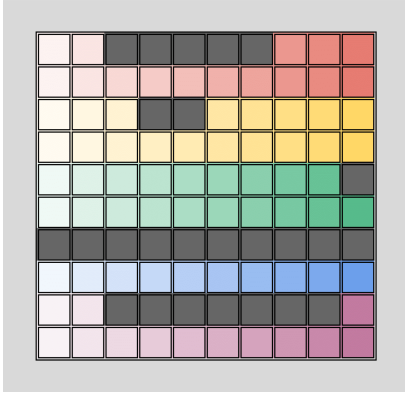
輸出

```
10
combine 1 2
cut 1 1
cut 1 2
combine 3 98
combine 31 41
cut 1 3
combine 99 100
cut 1 99
cut 1 1
spread
```

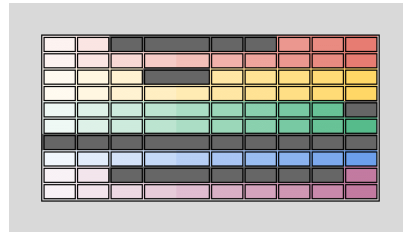
最佳解的 $q = 3$ 。如果你的輸出長這樣，將只能在子任務 3, 4 拿到部份分。

範例說明

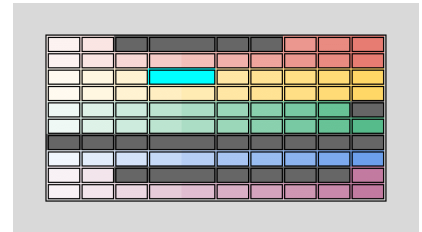
範例 1 圖示如下，其中灰色區塊表示所有要剪下來的布料、天藍色區塊為當前 **cut** 操作剪下的布料、亮綠色區塊為已經被剪下來的布料們、而分隔線消失代表著對應的 column 經由 **combine** 操作被壓縮在一起。



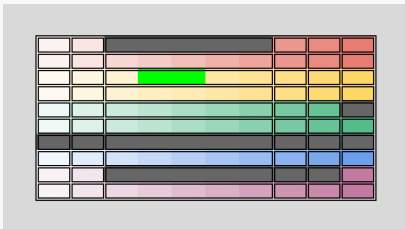
範例 1 的初始狀態



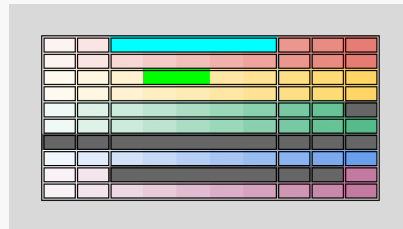
combine 4 5



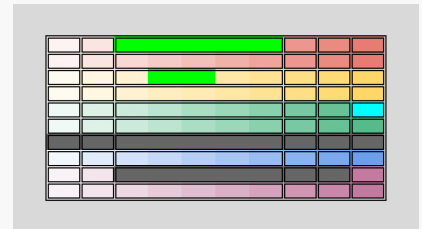
cut 3 5



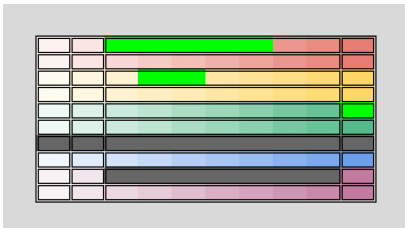
combine 3 7



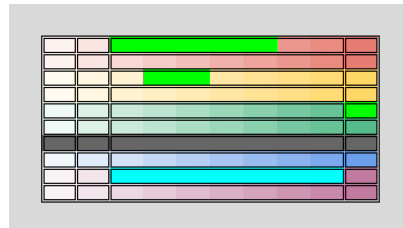
cut 1 5



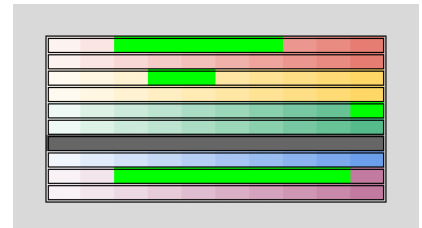
cut 5 10



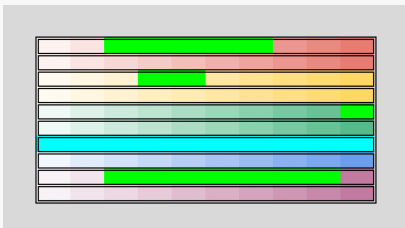
combine 6 9



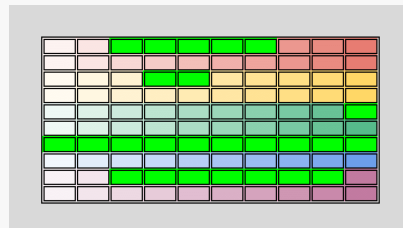
cut 9 8



combine 1 10



cut 7 1



spread

對於第六次的 **combine 6 9** 操作，由於該操作之前 column 3 到 7 已經被壓縮到一起了，因此該操作相當於 **combine 3 9**。

請記得你的最後一個操作**必須**是 **spread**。

範例 2 圖示如下，其中灰色區塊表示所有要剪下來的布料：



即便是被指定了多次的位置（如 $(1, 3)$ ），你也只需要對這個位置執行一次 **cut** 操作就好。

範例 3 中，即便沒有必要，你還是可以重複剪下同一個位置，也可以重複壓縮已經被壓縮過的區間。

提示：子任務 4 的部份分數比子任務 3 更好拿。

後記

多虧了你的幫忙，Nachoneko 媽媽成功的為孩子們做出了最適合的衣服，P 喵也如願以償的穿上了女裝。真是可喜可賀！