

B. 你好我的世界

Problem ID: HelloMinecraft

Time Limit: 1.0s

Memory Limit: 512MiB

今年正好是「我的世界 (Minecraft)」從開始開發至今的 15 周年，身為「我的世界」的忠實老玩家與一名新手的遊戲開發者，Samuel 開發的第一款遊戲名為「你好我的世界」，靈感正是來自於原版的「我的世界」。

這個遊戲的主要目標就是在世界中合成各種不同的物品。

當然了，還要有蓋房子、打怪、PVP、跑酷等等，只不過對於 Samuel 這個遊戲開發新手，這些功能還太複雜了，可能在看不見的將來才會加入遊戲中。

回到正題，在這款遊戲中有 n 種不同物品可以合成，編號為 $1, 2, \dots, n$ ，而除此之外還有 m 種不同的功能方塊可以製造，編號為 $1, 2, \dots, m$ 。

所謂功能方塊是指用於輔助合成物品，但不會被消耗的道具，也就是說取得一次就可以無限次使用了（對應原版我的世界遊戲中的合成台、熔爐等）。

為了合成各種不同的物品，你好我的世界遊戲中對於每種物品 i 皆有 way_i 種的合成方式，對於物品 i 的第 j 種合成方式會需要使用 $a_{i,j}$ 種物品與 $b_{i,j}$ 種功能方塊並花費 $c_{i,j}$ 秒。更嚴謹的說有 $a_{i,j}$ 組需求 (x_k, y_k) ，代表會消耗 y_1 個物品 x_1 、 y_2 個物品 x_2 、 \dots 、 $y_{a_{i,j}}$ 個物品 $x_{a_{i,j}}$ 來製造 1 個物品 i ，並且同時需要有 $b_{i,j}$ 種指定的功能方塊 $z_1, z_2, \dots, z_{b_{i,j}}$ 。

再次強調功能方塊並不會消耗。

請注意每種合成方式的 $a_{i,j}, b_{i,j}, c_{i,j}$ 皆可能為 0， $a = 0$ 代表不會消耗其他物品， $b = 0$ 代表不需要任何功能方塊， $c = 0$ 代表該次合成不需要時間。

另外功能方塊的取得方法，雖然正常來說也必須透過合成，但為了簡化遊戲，在早期版本直接設定可以花費 t_i 即可取得功能方塊 i 。

現在，身為 speedrunner 的你想要挑戰用最快的速度達到遊戲中的里程碑，因此你想知道對於每一種物品，理論上最快可以取得一件該物品的時間為何？

請注意，如果該物品無論如何都無法合成的話，請輸出 -1 。另外如果最小取得時間大於等於 10^{15} 秒的話，由於沒有那麼多時間玩遊戲，因此也視為無法拿到，輸出 -1 。

— 輸入 —

輸入的第一行包含兩個正整數 n, m ，分別代表物品的種類數、功能方塊的種類數。

第二行包含 n 個正整數 $way_1, way_2, \dots, way_n$ ，代表每種物品的合成方法數。

第三行包含 m 個非負整數 t_1, t_2, \dots, t_m ，代表每種功能方塊的取得所需時間。注意由於 m 可能為 0，因此該行可能為空。

接下來對於每種物品 i 有 way_i 種物品的合成方式，每種合成方式包含三行。其中的第一行的開頭有一個非負整數 $a_{i,j}$ ，接著有 $a_{i,j}$ 組正整數 $x_{i,j,k} y_{i,j,k}$ ，代表消耗的物品種類編號、數量；其中的第二行的開頭有一個非負整數 $b_{i,j}$ ，接著有 $b_{i,j}$ 個正整數 $z_{i,j,k}$ ，代表需要的功能方塊編號；其中的第三行有一個非負整數 $c_{i,j}$ ，代表此種合成方式需要的時間。

簡而言之，輸入格式如下。

```

n m
way_1 way_2 ... way_n
t_1 t_2 ... t_m
a_{1,1} x_{1,1,1} y_{1,1,1} x_{1,1,2} y_{1,1,2} ... x_{1,1,a_{1,1}} y_{1,1,a_{1,1}}
b_{1,1} z_{1,1,1} z_{1,1,2} ... z_{1,1,b_{1,1}}
c_{1,1}
...
a_{1,way_1} x_{1,way_1,1} y_{1,way_1,1} x_{1,way_1,2} y_{1,way_1,2} ... x_{1,way_1,a_{1,way_1}} y_{1,way_1,a_{1,way_1}}
b_{1,way_1} z_{1,way_1,1} z_{1,way_1,2} ... z_{1,way_1,b_{1,way_1}}
c_{1,way_1}
a_{2,1} x_{2,1,1} y_{2,1,1} x_{2,1,2} y_{2,1,2} ... x_{2,1,a_{2,1}} y_{2,1,a_{2,1}}
b_{2,1} z_{2,1,1} z_{2,1,2} ... z_{2,1,b_{2,1}}
c_{2,1}
...
a_{2,way_2} x_{2,way_2,1} y_{2,way_2,1} x_{2,way_2,2} y_{2,way_2,2} ... x_{2,way_2,a_{2,way_2}} y_{2,way_2,a_{2,way_2}}
b_{2,way_2} z_{2,way_2,1} z_{2,way_2,2} ... z_{2,way_2,b_{2,way_2}}
c_{2,way_2}
...
a_{n,1} x_{n,1,1} y_{n,1,1} x_{n,1,2} y_{n,1,2} ... x_{n,1,a_{n,1}} y_{n,1,a_{n,1}}
b_{n,1} z_{n,1,1} z_{n,1,2} ... z_{n,1,b_{n,1}}
c_{n,1}
...
a_{n,way_n} x_{n,way_n,1} y_{n,way_n,1} x_{n,way_n,2} y_{n,way_n,2} ... x_{n,way_n,a_{n,way_n}} y_{n,way_n,a_{n,way_n}}
b_{n,way_n} z_{n,way_n,1} z_{n,way_n,2} ... z_{n,way_n,b_{n,way_n}}
c_{n,way_n}

```

— 輸出 —

輸出 n 行，每行包含一個非負整數，第 i 行代表對於物品 i ，取得 1 個所需的最短時間，對於無法合成或所需最短時間大於等於 10^{15} 的物品輸出 -1 。

— 輸入限制 —

- $1 \leq n \leq 100$
- $0 \leq m \leq 10$
- $1 \leq way_i \leq 10$
- $0 \leq t_i \leq 10^9$
- $0 \leq a_{i,j} \leq \min(n, 10)$
- $0 \leq b_{i,j} \leq m$
- $0 \leq c_{i,j} \leq 100$
- $1 \leq x_{i,j,k} \leq n$
- $1 \leq y_{i,j,k} \leq 10$
- $1 \leq z_{i,j,k} \leq m$
- 對於任意 $d \neq e$ 有 $x_{i,j,d} \neq x_{i,j,e}$ 與 $z_{i,j,d} \neq z_{i,j,e}$

— 子任務 —

編號	分數	額外限制
1	0	範例輸入輸出
2	6	所有 $a_{i,j} = b_{i,j} = 0$
3	10	所有 $a_{i,j} = 0$
4	10	所有 $a_{i,j} = 1, b_{i,j} = 0$
5	14	所有 $a_{i,j} = 1$
6	20	所有 $way_i = 1$
7	40	無額外限制

— 範例輸入 1 —

```
3 0
2 2 1

0
0
5
1 2 1
0
1
0
0
2
1 1 2
0
0
2 1 2 2 2
0
10
```

— 範例輸出 1 —

```
3
2
20
```

— 範例輸入 2 —

```
3 2
2 2 1
15 20
0
2 1 2
5
1 2 1
0
1
0
0
```

2

1 1 2

1 2

0

2 1 2 2 2

2 2 1

10

— 範例輸出 2 —

3

2

55