A. 礦砂採集

問題描述

史蒂夫是一名礦場探險家，這天他到了一座新的礦場探險並發現了多種的礦砂。不幸地在花了好大的力氣挖了 n 種礦砂後，史蒂夫才發現帶來的背包能承受的重量上限為 M，可能無法帶回挖到的所有礦砂。史蒂夫測量礦砂 i 挖到的重量 w_i，並利用過去買賣礦砂的經驗推出礦砂 i 的單位重量價格 p_i。對於礦砂 i，若史蒂夫決定只帶回 x_i 單位重量，則可賣得 p_ix_i 的價格；此時帶回的礦砂的總價為 $p_1x_1 + p_2x_2 + \ldots + p_nx_n$。當然 x_i 必須滿足 $0 \le x_i \le w_i$ 且 $x_1 + x_2 + \ldots + x_n \le M$。

聰明的史蒂夫發現帶回礦砂的方式不只一種，且總價可能不同。舉例來說，挖到的礦砂有 8 種，且重量與單價如下：

礦砂編號 (i)	1	2	3	4	5	6	7	8
重量 (w_i)	2	3	12	5	3	2	4	10
單位重量價格 (p_i)	31	23	17	11	7	5	3	1

假設背包限重 $M = 6$。若史蒂夫選擇帶回 3 單位的礦砂 2 和 3 單位的礦砂 5，則礦砂總價為 $3 \times 23 + 3 \times 7 = 90$；若史蒂夫選擇帶回 2 單位的礦砂 1、3 單位的礦砂 2、以及 1 單位的礦砂 3，則礦砂總價為 $2 \times 31 + 3 \times 23 + 1 \times 17 = 148$，且 148 恰為背包限重 6 時能帶回的最大礦砂總價。

請撰寫一支程式告訴史蒂夫他能帶回的礦砂的總價最大值 P。

輸入格式

$$
\begin{array}{l}
n \;\; M \\
w_1 \;\; p_1 \\
w_2 \;\; p_2 \\
\vdots \\
w_n \;\; p_n
\end{array}
$$

- n 與 M 分別代表挖到的礦砂種數和背包的限重。
- w_i 與 p_i 分別代表礦砂 i 挖到的重量和單位重量的價格。

輸出格式

P

- P 為一個整數，代表能帶回的礦砂總價最大值。

測資限制

- $1 \leq n \leq 1000$。
- $1 \leq M \leq 100000$。
- $1 \leq w_i \leq 100$。（$i \in \{1, 2, \ldots, n\}$）
- $1 \leq p_i \leq 1000$。（$i \in \{1, 2, \ldots, n\}$）
- 輸入的數皆為整數。

範例測試

Sample Input	Sample Output
5 3 1 11 2 7 1 5 2 3 1 2	25
8 13 9 5 1 7 2 9 5 8 5 2 7 10 3 2 6 3	120

評分說明

本題共有二組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	50	輸入滿足 $p_1 \geq p_2 \geq \ldots \geq p_n$。
2	50	無額外限制。

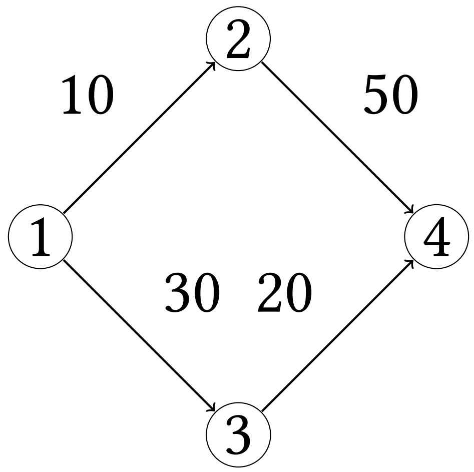
B. 村莊與小徑

問題描述

某小鎮由 n 個村莊及 m 條連接兩個不同的村莊的小徑組成。這 n 個村莊分別以正整數 $\{1, 2, \cdots, n\}$ 來編號。某天鎮上的議會通過了推廣觀光的法案，作為推廣觀光的第一步，鎮長決定規劃一些觀光路徑給觀光客參觀；為了讓遊客能有效率的參觀鎮上的景點，這些路徑會滿足某些規則。首先，鎮長決定讓所有小徑變成單行道只允許單一方向通行來保持動線流暢避免壅塞；再來，實施單行道政策後必須保證鎮上不會出現環狀路徑——也就是保證不會有從某一個村莊出發經過某些單向的小徑再回到原村莊的情形出現；鎮長認為這可以加強觀光效率並避免觀光客迷路。

目前鎮內已經決定所有小徑的通行方向使得鎮中不存在環狀路徑；為了保護觀光客安全，他們還聘請專家評估了每條小徑的**危險程度評分**，此評分可能是正的也可能是負的。所有來小鎮觀光的人都會從村莊 1 出發，循著某些小徑走到某個村莊後離開。現在的工作是希望對每一個編號 1 以外的村莊 i 規劃出一條最佳路徑 P_i，所謂的最佳路徑是指「該路徑所經過小徑的危險評分總和是所有從 1 走到 i 的路徑中最小的」，我們將這個危險度總合稱為「到村莊 i 的危險評分」。請寫一支程式計算出每個村莊的最佳路徑，並輸出所有村莊危險評分的總和。出發位置村莊 1 的危險評分定義評分為 0，不需要另外計算。

例如下面例子：



村莊 2，3，4 的危險評分分別為 10，30 與 50，因此 $10 + 30 + 50 = 90$ 即為所求。

輸入格式

$n\ \ m$
$u_1\ \ v_1\ \ w_1$
$u_2\ \ v_2\ \ w_2$
$\vdots$
$u_m\ \ v_m\ \ w_m$

- n，m 代表村莊數量以及小徑數量。
- u_i，v_i，w_i 代表第 i 條路徑是由村莊 u_i 通往村莊 v_i 且危險評分 w_i 的單行道。

輸出格式

$answer$

- $answer$ 為一整數，代表所有村莊最佳路徑的危險評分總和。

測資限制

- $2 \leq n \leq 100000 = 10^5$。
- $1 \leq m \leq 200000 = 2 \times 10^5$。
- $1 \leq u_i \leq n$。（$i \in \{1, 2, \ldots, m\}$）
- $1 \leq v_i \leq n$。（$i \in \{1, 2, \ldots, m\}$）
- $|w_i| \leq 110000$。（$i \in \{1, 2, \ldots, m\}$）
- 輸入的圖沒有環（從任意村出發無法回到該起點村莊），且村莊 1 可以經由單向小徑走到其他所有村莊。
- 輸入所有數字皆為整數。

範例測試

Sample Input	Sample Output
4 4 1 2 10 2 4 50 1 3 30 3 4 20	90
5 6 1 3 10 3 4 0 1 5 30 5 4 10 3 2 10 3 4 40	70

評分說明

本題共有二組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	32	$n \leq 100$ 且 $m \leq 1000$。
2	68	無額外限制。

C. 樣本解析

問題描述

湯姆是一位癌症專家，他最近得到了一組腫瘤樣本，發現其中有一個樣本組成特別奇怪；他在嘗試解析此樣本時遇到了以下難題，希望你能寫一個程式幫忙解決。

一個樣本可以用一個非空集合表示，集合中可能出現的元素不超過 26 個，不同的元素各以一個小寫英文字母表示。給定兩個的樣本 A 和 B，湯姆將樣本之間的關係分成以下四種：

1. **互無交集**：A 和 B 沒有任何共同的元素。
2. A **真包含** B：A 包含了 B 中所有元素且兩集合不相等。意即每個出現在 B 裡的元素亦出現在 A 集合中，且 A 裡至少有一個元素未在 B 集合出現。
3. B **真包含** A：B 包含了 A 中所有元素且兩集合不相等。意即每個出現在 A 裡的元素亦出現在 B 集合中，且 B 裡至少有一個元素未在 A 集合出現。
4. **部分交集**：A 和 B 兩集合不相等，且不符合上述三個條件。

例如，若樣本 $A = \{b, e\}, B = \{a, b, e\}, C = \{a, c, g\}$，則我們說 B 真包含 A、A 和 C 互無交集、B 和 C 部分交集。

湯姆認為正常的樣本之間不應該有部分交集的關係，他目前蒐集的所有樣本裡包含 n 個不同的正常樣本 $S_1, S_2, \cdots, S_n$ 和一個異常的樣本 X，正常的樣本間兩兩沒有部份交集的關係，而異常樣本 X 和至少一個正常樣本有部分交集的關係。湯姆想知道下面五個參數：

- P_1：有多少正常樣本 S_i 和 X 有「互無交集」的關係。
- P_2：有多少正常樣本 S_i 和 X 有「S_i 真包含 X」的關係。
- P_3：有多少正常樣本 S_i 和 X 有「X 真包含 S_i」的關係。
- P_4：有多少正常樣本 S_i 和 X 有「部分交集」的關係。
- P_5：有多少方法能將 X 拆分成兩個子樣本 X_1 和 X_2，使得 X 中每個元素正好在其中一個子樣本中，且 $X_1, X_2, S_1, S_2, \cdots, S_n$ 共 $n + 2$ 個樣本間兩兩沒有部分交集的關係。兩種方法不同若且唯若拆出的第一個子樣本 X_1 不同。

例如，$n = 3, S_1 = \{a, b, e\}, S_2 = \{a, b\}, S_3 = \{c, d\}$，若 $X = \{a, b, f\}$，則可將 X 拆成 $X_1 = \{a, b\}$ 和 $X_2 = \{f\}$，這樣一來 X_1, X_2, S_1, S_2, S_3 之間便找不到有部分交集的兩集合。這個例子的另一種可行拆解方法是取 $X_1 = \{f\}$ 且 $X_2 = \{a, b\}$。若 $X = \{a, b, c, f\}$，則沒有任何方法可以將 X 拆成兩個子樣本滿足上述條件。

給定異常樣本 X 和 n 個不同的正常樣本 $S_1, S_2, \cdots, S_n$，請寫一個程式幫助湯姆解決以上問題。

輸入格式

n
X
S_1
S_2
$\vdots$
S_n

- n 代表正常樣本數量。
- X 為湯姆收到的異常樣本。
- S_i 為第 i 個湯姆收到的正常樣本。
- X 與 S_i 都以一個字串表示，字串裡的各字元分別代表一個集合裡的元素。

輸出格式

P_1 P_2 P_3 P_4 P_5

- P_1 到 P_5 為五個整數，代表題目所求的答案。

測資限制

- $1 \leq n \leq 18$，且 n 為整數。
- X 的長度至多為 20，且至少由一個字母組成。
- S_i 的長度至多為 20，且至少由一個字母組成。($i \in \{1, 2, \ldots, n\}$)
- X 與 S_i 都僅由不重複的英文小寫字母組成。($i \in \{1, 2, \ldots, n\}$)
- 正常樣本間沒有「部分交集」的關係，且兩兩互不相等。
- X 至少和某個正常樣本 S_i 有部分交集的關係。($i \in \{1, 2, \ldots, n\}$)

範例測試

Sample Input	Sample Output

`1`
`acde`
`bcda`
	`0 0 0 1 2`

`3`
`abf`
`eab`
`ab`
`cd`
	`1 0 1 1 2`

`3`
`ace`
`ab`
`cd`
`ef`
	`0 0 0 3 0`

評分說明

本題共有三組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	23	所有集合的元素都由前 10 個小寫英文字母組成 (a ~ j) 且 $P_5 = 0$。
2	50	所有集合的元素都由前 10 個小寫英文字母組成 (a ~ j)。
3	27	無額外限制。

D. 水果包裝

問題描述

阿明暑假期間在一間水果包裝工廠打工，這間工廠有一台自動化包裝機協助阿明將水果分配包裝以送往賣場販售。每次進行包裝前阿明會將 n 顆水果以某個順序排成一列在輸送帶上，並放入編號 $1, 2, \ldots, m$ 共 m 個空袋子至包裝機內，最後再啟動輸送帶將水果送至包裝機包裝。包裝機在輸送帶啟動後，會依照小明擺在輸送帶上的順序將水果一顆一顆放入袋子裡。每一步機器會將輸送帶上最前面的水果放入重量最小的袋子裡；如果有多個重量最小的袋子，則機器會選擇編號最小的袋子放入。

有一天阿明發現某一批袋裝水果的重量差異有些懸殊，懷疑包裝機並沒有正常運作。雖然阿明已經忘記當初水果放在輸送帶上的順序了，但他還是想請你幫他看看這個包裝結果是有沒有可能出現的，阿明將這批袋裝水果中的每顆水果編號為 $1, 2, \ldots, n$，並給了你這些水果的重量 $w_1, w_2, \ldots, w_n$ 以及最後的包裝結果。請你幫阿明求出一個 $\{1, 2, \ldots, n\}$ 的排列 σ，使得當包裝機正常運作時，只要收到的水果編號依序為 $\sigma(1), \sigma(2), \ldots, \sigma(n)$，就會得到他給你的包裝結果。

輸入格式

$$
\begin{array}{l}
n\ \ m \\
w_1\ \ w_2\ \ \cdots\ \ w_n \\
k_1\ \ p_{1,1}\ \ p_{1,2}\ \ \cdots\ \ p_{1,k_1} \\
k_2\ \ p_{2,1}\ \ p_{2,2}\ \ \cdots\ \ p_{2,k_2} \\
\vdots \\
k_m\ \ p_{m,1}\ \ p_{m,2}\ \ \cdots\ \ p_{m,k_m}
\end{array}
$$

- n 與 m 分別代表水果的個數和分裝的袋數。
- w_i 代表水果 i 的重量。
- k_i 代表袋子 i 裝的水果數量。
- $\{p_{i,1}, p_{i,2}, \ldots, p_{i,k_i}\}$ 代表袋子 i 裝的水果集合。

輸出格式

若存在符合條件的排列 σ，請輸出

$$\sigma(1)\ \ \sigma(2)\ \ \cdots\ \ \sigma(n)$$

其中 $\sigma(1)$ 到 $\sigma(n)$ 必須是一個 1 到 n 的排列，否則輸出

$$-1$$

代表不存在題目所求的排列。注意，如果有複數種可能的排列，輸出任意一種皆可。

測資限制

- $1 \leq n \leq 200000 = 2 \times 10^5$。
- $1 \leq m \leq n$。
- $1 \leq w_i \leq 10^9$。($i \in \{1, 2, \ldots, n\}$)
- $0 \leq k_i \leq n$ 且 $\sum_{i=1}^{m} k_i = n$。
- $1 \leq p_{i,j} \leq n$。($i \in \{1, 2, \ldots, m\}$，$j \in \{1, 2, \ldots, k_i\}$)
- 所有水果均已被裝袋，亦即對於任意介於 1 和 n 之間的正整數 q，均存在唯一的數對 (i, j) 使得 $p_{i,j} = q$。($i \in \{1, 2, \ldots, m\}$，$j \in \{1, 2, \ldots, k_i\}$)
- 輸入的數皆為整數。

範例測試

Sample Input	Sample Output
3 2 1 4 6 2 1 2 1 3	1 3 2
4 2 1 4 5 6 2 1 3 2 4 2	1 2 3 4
4 2 1 4 2 2 3 1 3 2 1 4	-1

評分說明

本題共有四組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	5	$n \le 9$ 且 $w_i \le 100$。
2	23	$n \le 18$ 且 $w_i \le 100$。
3	57	$n \le 100$ 且 $w_i \le 100$。
4	15	無額外限制。

E. 共同朋友

問題描述

共同朋友的數量是社會網路分析中評估兩人關係的一個重要指標。現有一個研究樣本的社會網路是由 n 個人組成的，這些人分別以正整數 1 到 n 來編號。已知第 i 個人的朋友有 d_i 個，並且朋友關係皆為對稱，也就是說，若 a 為 b 的朋友，則 b 也必為 a 的朋友。對於某相異兩人 a 與 b，若另一人 c 同時是 a 與 b 的好友，則稱 c 為兩人的共同朋友。本題要計算有多少對 (a, b) 滿足 $a < b$ 且兩人至少有一位共同朋友。

以下是一個 $n = 4$ 的例子。

- 1 的朋友清單：$\{2, 3\}$。
- 2 的朋友清單：$\{1, 3\}$。
- 3 的朋友清單：$\{2, 1, 4\}$。
- 4 的朋友清單：$\{3\}$。

根據定義 $(2, 3)$ 有一位共同朋友 1 為題目所求的數對。另外 $(1, 2), (1, 3), (1, 4), (2, 4)$ 也都是題目所求，因此答案為 5。

輸入格式

$$
\begin{array}{l}
n \\
d_1 \;\; f_{1,1} \quad f_{1,2} \;\; \cdots \;\; f_{1,d_1} \\
d_2 \;\; f_{2,1} \quad f_{2,2} \;\; \cdots \;\; f_{2,d_2} \\
\vdots \\
d_n \;\; f_{n,1} \quad f_{n,2} \;\; \cdots \;\; f_{n,d_n}
\end{array}
$$

- n 為總人數。
- 第 i 個人朋友數量為 d_i。
- 第 i 個人的第 j 個朋友為 $f_{i,j}$。

輸出格式

```
answer
```

- $answer$ 為一整數，代表有多少對 (a, b)，滿足 $a < b$ 且兩人至少有一位共同朋友。

測資限制

- $1 \leq n \leq 2500$。
- $0 \leq d_i \leq n - 1$。($i \in \{1, 2, \ldots, n\}$)
- $0 \leq f_{i,j} \leq n - 1$。($i \in \{1, 2, \ldots, n\}$，$j \in \{1, 2, \ldots, d_i\}$)
- $n, k, d_i, f_{i,j}$ 皆為整數。($i \in \{1, 2, \ldots, n\}$，$j \in \{1, 2, \ldots, d_i\}$)
- 此份朋友清單一定滿足朋友關係皆為對稱。
- 自己不會是自己的朋友。
- 同一個人的朋友清單不會有重複的人。

範例測試

Sample Input	Sample Output
6 4 6 2 3 5 2 5 1 3 6 1 5 0 4 6 2 3 1 3 1 3 5	10
4 2 2 3 2 1 3 3 2 1 4 1 3	5

評分說明

本題共有三組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	9	$n \leq 200$。
2	12	每個人的朋友人數皆不超過 200。
3	79	無額外限制。

F. 歡樂外送點

問題描述

矩形市中所有的道路都是南北或東西走向，這些道路不但每一條都貫穿矩形市的範圍，而且每一條相鄰南北 (與東西) 向的道路都是間隔 1 單位距離。

矩形市的規模非常龐大，我們用南北 x 路與東西 y 路來表示各個道路的名稱，其中 x 與 y 可以是 -2×10^8 到 2×10^8 之間的任意整數。南北 x 路和東西 y 路所形成的交叉路口可以用平面上的一對整數點座標 (x, y) 來表示。而且對於任兩個交叉路口座標 (x_1, y_1) 與 (x_2, y_2)，他們之間的路程距離為 $|x_1 - x_2| + |y_1 - y_2|$。

四十五度歡樂購是一家購物服務公司，該公司在矩形市有 n 家提供購物服務的商店，每家店都開在交叉路口上。其中第 i 家商店的位置在路口 (x_i, y_i) 且該商店的服務半徑為 R_i，代表商店最遠的服務距離。也就是說，任何路口與第 i 家商店的距離不超過 R_i 商店才會提供服務，反之就會被商店視為距離過遙遠而拒絕提供服務。請注意，某些商店的服務半徑為 0，表示只有商店所在的路口可以得到該商店的服務。

已知第 i 家商店有一個繁榮指數 W_i，用來表示路口得到此店服務所增加的繁榮度，且一個路口的繁榮度定義為所有可以服務此路口的商店的繁榮度指數總和。請你寫一支程式計算出，在矩形市內所有路口繁榮度的最大值。

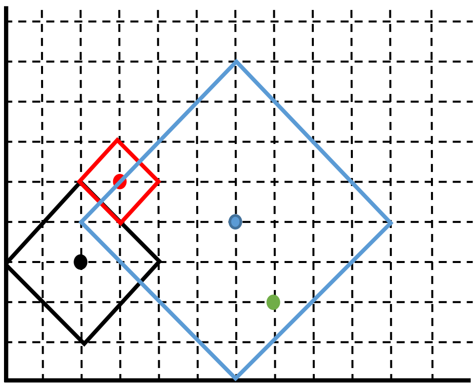


圖 1

上圖是一個例子，在本例中有四家商店，分別是：

- $(x_1, y_1) = (2, 3)$、$R_1 = 2$、$W_1 = 10$。
- $(x_2, y_2) = (3, 5)$、$R_2 = 1$、$W_2 = 20$。
- $(x_3, y_3) = (6, 4)$、$R_3 = 4$、$W_3 = 30$。
- $(x_4, y_4) = (7, 2)$、$R_4 = 0$、$W_4 = 40$。

其位置與範圍如圖中所標示，舉例來說：

- 在路口 $(2,4)$ 可以得到第 1 家與第 3 家商店的服務，繁榮度為 $10+30=40$。
- 在路口 $(3,4)$ 可以得到第 1、2、3 家商店的服務，繁榮度為 $10+20+30=60$。
- 在路口 $(7,2)$ 可以得到第 3 與第 4 家商店的服務，繁榮度為 $30+40=70$。

此例中繁榮度最大的路口是路口 $(7,2)$，你要輸出為它的繁榮度 70。

輸入格式

n
$x_1 \;\; y_1 \;\; R_1 \;\; W_1$
$x_2 \;\; y_2 \;\; R_2 \;\; W_2$
$x_3 \;\; y_3 \;\; R_3 \;\; W_3$
$\vdots$
$x_n \;\; y_n \;\; R_n \;\; W_n$

- n 代表商店數。
- 商店 i 所在的交叉路口 x 座標為 x_i。
- 商店 i 所在的交叉路口 y 座標為 y_i。
- 商店 i 的服務半徑為 R_i。
- 商店 i 的繁榮指數為 W_i。

輸出格式

$answer$

- $answer$ 為一整數，代表矩形市內所有路口繁榮度的最大值。

測資限制

- $1 \leq n \leq 3 \times 10^5$。
- $0 \leq x_i \leq 10^8$。($i \in \{1,2,\ldots,n\}$)
- $0 \leq y_i \leq 10^8$。($i \in \{1,2,\ldots,n\}$)
- $0 \leq R_i \leq 10^8$。($i \in \{1,2,\ldots,n\}$)
- $1 \leq W_i \leq 100$。($i \in \{1,2,\ldots,n\}$)
- n, x_i, y_i, R_i, W_i 皆為整數。($i \in \{1,2,\ldots,n\}$)

範例測試

Sample Input	Sample Output
4 6 0 1 2 1 0 4 2 4 0 1 3 2 0 0 3	7
4 2 3 2 10 3 5 1 20 6 4 4 30 7 2 0 40	70

評分說明

本題共有三組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	13	$n \leq 3000$。
2	21	$n \leq 5 \times 10^4$。
3	66	無額外限制。

G. 矩陣相乘

問題描述

某科學運算中心打算重新打造組合最佳化領域的函式庫，其中一項任務為撰寫一個在模運算 (modular arithmetic) 下的矩陣相乘函式。此任務中的所有加法與乘法都在模 P 下定義（其中 P 為一個質數），亦即兩整數相加／相乘的結果為原始運算結果除以 P 的餘數。另，此任務考慮的矩陣是一個 n 列 n 行的二維數列，用 $A_{i,j}$ 來表示矩陣 A 在第 i 列第 j 行（簡稱 (i,j) 元素）的值。例如

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

則我們有 $A_{1,2} = 2$ 且 $A_{3,2} = 8$。

給定兩個 $n \times n$ 矩陣 A 和 B, 定義相乘結果 AB 為另一個 $n \times n$ 矩陣 C, 其中 $C_{i,j} \equiv \sum_{k=1}^{n} A_{i,k} B_{k,j} \pmod{P}$。舉例來說，設

$$A = \begin{bmatrix} 1402 & 1400 \\ 4 & 2799 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

在模 $P = 2801$ 下的相乘結果 C 為

$$C = AB = \begin{bmatrix} 1402 & 1400 \\ 4 & 2799 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 2799 & 0 \end{bmatrix}$$

其中

- $C_{1,1} = 1402 \cdot 1 + 1400 \cdot 3 \equiv 0 \pmod{2801}$
- $C_{1,2} = 1402 \cdot 2 + 1400 \cdot 4 \equiv 1 \pmod{2801}$
- $C_{2,1} = 4 \cdot 1 + 2799 \cdot 3 \equiv 2799 \pmod{2801}$
- $C_{2,2} = 4 \cdot 2 + 2799 \cdot 4 \equiv 0 \pmod{2801}$

透過統計分析得知，組合最佳化領域中的矩陣相乘結果常常是稀疏矩陣（大部份元素為 0 的矩陣）；本題的目標即為在矩陣乘積稀疏的條件下，快速算出兩個矩陣相乘的結果。

給定兩個矩陣 A, B 以及模數 P，已知 A, B 相乘會得到一個非 0 元素個數不超過 $2n$ 的稀疏矩陣 C，請列出 C 的每個非 0 元素的位置及其值。

注意：在實作過程中**請特別留意 mod 運算的次數**, 例如在計算 $C_{i,j}$ 時可以用 64 位元的整數儲存避免溢位，並於加總後做一次 mod 運算即可。本題的時限是以程式中模運算的次數為 $O(n^2)$ 或以下作為前提設定的，如果超過這個次數很有可能會造成演算法正確卻超時的結果。

輸入格式

$$
\begin{array}{l}
subtask \\
n \;\; P \\
A_{1,1} \;\; A_{1,2} \;\; \cdots \;\; A_{1,n} \\
A_{2,1} \;\; A_{2,2} \;\; \cdots \;\; A_{2,n} \\
\vdots \\
A_{n,1} \;\; A_{n,2} \;\; \cdots \;\; A_{n,n} \\
B_{1,1} \;\; B_{1,2} \;\; \cdots \;\; B_{1,n} \\
B_{2,1} \;\; B_{2,2} \;\; \cdots \;\; B_{2,n} \\
\vdots \\
B_{n,1} \;\; B_{n,2} \;\; \cdots \;\; B_{n,n}
\end{array}
$$

- $subtask$ 可能為 $1, 2, 3$，代表這筆測資所滿足的 $subtask$ 限制（詳見測資限制一節）。
- n 與 P 分別代表矩陣大小與矩陣運算所用的模數。
- $A_{i,j}$ 與 $B_{i,j}$ 分別代表 A 矩陣與 B 矩陣在 (i, j) 的值。

輸出格式

$$
\begin{array}{l}
x_1 \;\; y_1 \;\; v_1 \\
x_2 \;\; y_2 \;\; v_2 \\
\vdots \\
x_k \;\; y_k \;\; v_k
\end{array}
$$

- k 代表 A, B 的相乘結果 C 中，非 0 元素的個數。
- x_i y_i v_i 代表 $C_{x_i,y_i} = v_i$ 且 $v_i \neq 0$。
- 輸出的順序須按 (x_i, y_i) 排序，首先將 x_i 從小排到大，若 x_i 相同則按照 y_i 從小到大排序。

測資限制

- $subtask \in \{1, 2, 3\}$。
- $2 \le n \le 2800$。
- $37 \le P < 5 \times 10^7$，且 P 為質數。
- $0 \le A_{i,j} \le P - 1$。
- $0 \le B_{i,j} \le P - 1$。
- C 的非 0 元素個數介於 1 和 $2n$ 之間。
- 若 $subtask = 1$，保證 $P \ge 2801$，且 C 每列恰有 1 個非 0 元素。
- 若 $subtask = 2$，保證 $P \ge 2801$，且 C 每列恰有 2 個非 0 元素。
- 輸入的數皆為整數。

範例測試

Sample Input	Sample Output
1 2 2801 1402 1400 4 2799 1 2 3 4	1 2 1 2 1 2799
2 3 65537 1 1 2 65536 0 1 2 3 6 3 2 65536 65527 65532 1 4 2 0	1 1 1 1 2 1 2 1 1 2 3 1 3 2 1 3 3 1

評分說明

本題共有三組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	21	$subtask = 1$，亦即 $P \geq 2801$ 且 C 的每列恰有 1 個非 0 元素。
2	45	$subtask = 2$，亦即 $P \geq 2801$ 且 C 的每列恰有 2 個非 0 元素。
3	34	無額外限制。

H. 跑跑遊戲場

問題描述

（這題執行時間限制比其他題短很多，請特別注意）

某智慧機器鼠的研究機構為了測試機器鼠的行為表現，請你設計一座符合要求的機器鼠遊戲場以進行機器鼠的跑跑研究。遊戲場為一個 $N \times M$ 個方格所形成的空間，並且四周都用牆壁隔起來防止機器鼠跑到場外。我們用座標 (i, j) 來代表場內第 i 列第 j 行的格子。機器鼠從座標 $(1, 1)$ 的方格出發，目標是座標為 (N, M) 的方格, 如下圖 (1) 所示。

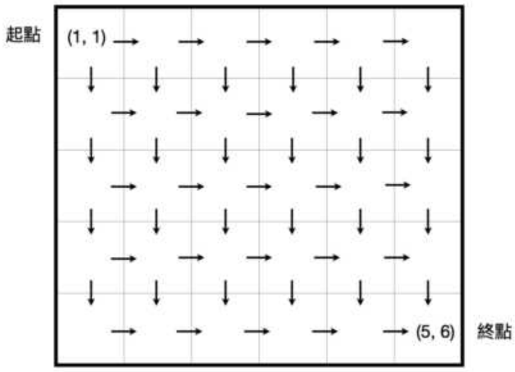


圖 1

遊戲場內兩個相鄰的方格之間都分別設有一道小拱門可以讓實驗者選擇把拱門關起來讓機器鼠無法通過；或是選擇把它打開，讓機器鼠可以通過。從起點開始，機器鼠每一步可以在場內往拱門開啟且相鄰的格子移動，由於機器鼠的設計很簡單，位於 (i, j) 時僅能往 $(i + 1, j)$ 或 $(i, j + 1)$ 前進，若這兩個格子都無法前進則會被卡在 (i, j) 無法動彈。現在研究機構希望能設計一個實驗讓機器鼠能順利從起點 $(1, 1)$ 通往終點 (N, M) 並算出走到終點的走法數。如下圖 (2) 所示，粗線代表沒法沒辦法通過的拱門，格子上的數字則代表起點到目前格子的相異走法數。

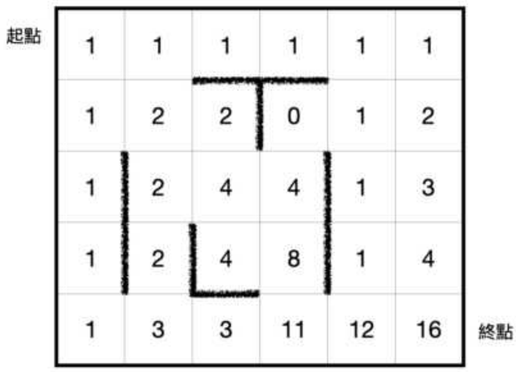


圖 2

研究人員給了你一個整數 T，你的目標是要設計遊戲場的大小（N 與 M）以及任意兩相鄰格之間的拱門開關狀態，使得機器鼠從 $(1,1)$ 到 (N,M) 之間恰好有 T 種不同的走法。此外，為了避免機器鼠卡住走不到終點，拱門的開關設計**不可以讓機器鼠會卡在某個非終點的格子**，也就是說任何從起點可以走到的格子都必須要能通向終點 (N,M)。研究人員請你設計一個滿足上述條件的遊戲場，並希望遊戲場周長越小越好。由於滿足要求的設計可能不只一種，你只需要輸出任何一種就可以了。

輸入格式

T

- T 代表研究人員希望從起點到終點不同走法的數量。

輸出格式

$N \ \ M$
K
$a_1 \ \ b_1 \ \ c_1 \ \ d_1$
$a_2 \ \ b_2 \ \ c_2 \ \ d_2$
$\vdots$
$a_K \ \ b_K \ \ c_K \ \ d_K$

- N，M 為兩個整數代表你設計的遊戲場的列數與行數。
- K 為一整數代表有幾道拱門需要關閉。
- a_i，b_i，c_i，d_i 為四個整數代表第 i 個需要關閉的拱門是位於格子 (a_i,b_i) 與 (c_i,d_i) 之間，其中 (a_i,b_i) 與 (c_i,d_i) 必須是存在且相鄰的兩格子。並且所有未輸出的拱門都視為開啟。
- 滿足題目敘述條件的任意一種地圖均視為合法輸出，但得分由特殊方法計算。關於分數計算方法請參照下面評分說明一節。

測資限制

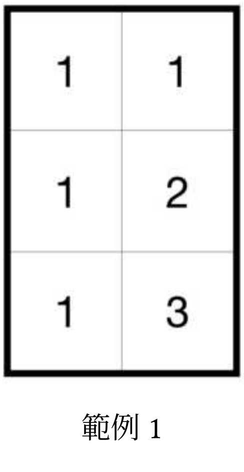
- $1 \le T \le 10^{18}$，且 T 為整數。

範例測試

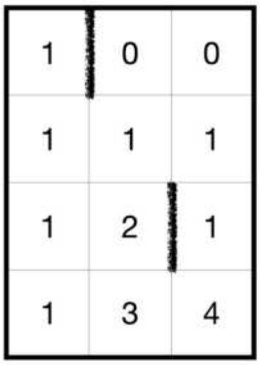
Sample Input	Sample Output
3	3 2 0
4	4 3 2 1 1 1 2 3 2 3 3
12	5 5 9 1 3 2 3 1 4 2 4 2 3 2 4 3 1 3 2 3 4 3 5 4 1 4 2 4 2 4 3 4 3 5 3 4 4 4 5

範例解釋

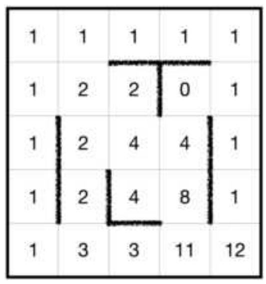
三筆測試資料的範例輸出示意圖如下：



範例 1



範例 2



範例 3

評分說明

輸出的遊戲場資料必須滿足下列要求：

1. 機器鼠走得到的所有方格都必須要可以通往終點。
2. $N + M \leq 140$。
3. 所輸出要關閉的拱門必須存在且不可以超出範圍或是非相鄰的格子，並且同一道拱門只可以輸出一次。
4. 起點到終點的走法必須恰為 T 種。

對每一筆測資，如果輸出不滿足以上任一要求則得分為 0 分；如果滿足要求且 $N + M \leq 90$，則為 100 分；否則，你的得分為 $(140 - N - M) \times 2$。本題有複數筆測資，整題的得分為各筆測資中得分之最小值。

I. 黑白機

問題描述

工廠有兩台計算機器黑機、白機以及一 n 個計算工作的序列。這些工作依序編號為 1 ~ n。這 n 個計算工作都可以在黑機或白機任一台進行運算，但所需時間不盡相同。我們以 b_i 與 w_i 分別代表第 i 項工作在黑機與白機進行運算所需要花費的時間。現在的任務是將每一件工作安排至黑機或是白機來執行，以盡早完成所有工作。

每一件工作運算完成後都會產生若干資料，而工作 i 必須取得編號 $1, 2, 3, \ldots, i-1$ 所有工作產生的資料之後才可以開始進行運算。將工作 i 所產生的資料從黑機送到白機或者從白機送到黑機都需要花費 t_i 的時間，要注意的是同一台機器**可以同時進行多筆資料的傳送加上一個計算工作**。假設工作 i 在 f_i 時在白機完成計算，其他在白機執行運算的工作都可以在 f_i 時就上取得工作 i 的資料；但如果是在黑機運算的工作則需要等到 $f_i + t_i$ 才能取得工作 i 的運算結果。已知工作 1 在時間 0 時開始進行，請計算出最早能完成工作 n 運算的時間。

以下是一個 $n = 4$ 的例子。假設 $b_1 = 2, b_2 = 2, b_3 = 3, b_4 = 5$；$w_1 = 1, w_2 = 1, w_3 = 7, w_4 = 2$；$t_1 = 4, t_2 = 1, t_3 = 1, t_4 = 1$。如果將工作 1, 2 與 4 安排在白機，工作 3 安排在黑機，則完成運算的時間是 11。

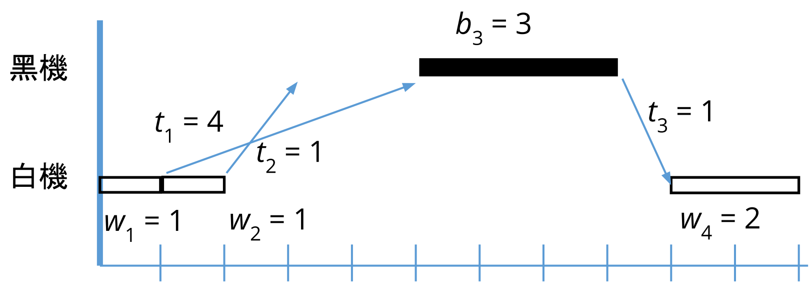


圖 1

如下圖，如果將工作 1, 2 改移到黑機，則完成時間為 10，這是上述例子最好的結果。

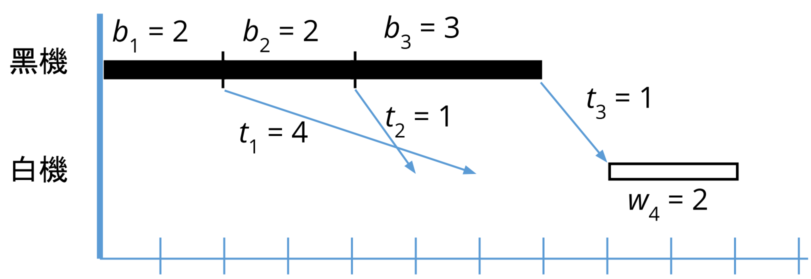


圖 2

輸入格式

n
$b_1 \ b_2 \ \cdots \ b_n$
$w_1 \ w_2 \ \cdots \ w_n$
$t_1 \ t_2 \ \cdots \ t_n$

- n 為工作個數。
- 工作 i 在黑機執行所需花費的時間為 b_i。
- 工作 i 在白機執行所需花費的時間為 w_i。
- 工作 i 在黑白機之間傳送產生的資料所需花費的時間為 t_i。

輸出格式

$answer$

- $answer$ 為一個整數，代表最早完成時間。

測資限制

- $1 \le n \le 10^5$。
- $1 \le b_i \le 10^4$。($i \in \{1, 2, \ldots, n\}$)
- $1 \le w_i \le 10^4$。($i \in \{1, 2, \ldots, n\}$)
- $1 \le t_i \le 10^4$。($i \in \{1, 2, \ldots, n\}$)
- n, b_i, w_i, t_i 皆為整數。($i \in \{1, 2, \ldots, n\}$)

範例測試

Sample Input	Sample Output
4 2 2 3 5 1 1 7 2 4 1 1 1	10
3 4 3 3 1 7 2 2 3 1	9
5 1 1 9 1 4 3 3 2 5 2 7 1 2 1 2	14

評分說明

本題共有三組子任務，條件限制如下所示。每一組可有一或多筆測試資料，該組所有測試資料皆需答對才會獲得該組分數。

子任務	分數	額外輸入限制
1	9	$n \leq 20$。
2	17	$n \leq 3000$。
3	74	無額外限制。