

103學年度高級中學資訊學科能力競賽決賽

上機程式設計競賽注意事項

- 一、本競賽採電腦線上自動評分，程式必須依規定上傳至評分主機。請嚴格遵守每一題目之原始程式格式、檔名、輸入方式及輸出方式等規定。若未遵守，該題將以 **0** 分計算。
- 二、本次競賽採取全面回饋機制。程式上傳至評分主機後，將自動編譯並進行測試。視等待評分題數多寡，該題測試結果及該題得分數將可於短時間內得知。程式可重複上傳及評分，但同一題兩次上傳之間必須間隔二分鐘以上。每題最終分數以該題所有單次評分結果之最高分計算。
- 三、本次競賽每一題皆有不同難易度的測試資料，詳細配分及限制條件請詳各題題目說明。程式執行時，針對每組測試資料執行時間**除題目有另外規定外**，以 **3** 秒為限。執行時間以評分主機為準。執行時間超過規定者，視同未完成，該組測試資料得分將以 **0** 分計算。每題可使用記憶體空間以 **512MB** 為限。
- 四、本次競賽學生需上傳原始程式碼 (.c, .cpp, .pas)。除題目另有規定外，輸入資料則自**標準輸入** (Standard input) 讀取，計算結果須輸出至**標準輸出** (Standard output)。注意：**請勿自行讀寫檔案**，若因此造成線上評分系統無法評分，該次評分結果將以 **0** 分計算。若藉此妨礙線上評分系統運行，承辦單位得取消比賽資格。
- 五、線上評分系統使用的編譯器與試場電腦所安裝的編譯器版本不一定一致，**以評分系統為主**。
- 六、請用承辦單位提供之**個人電腦內硬碟或網路儲存空間**備份原始程式碼。若由於任何原因需更換電腦時，僅能將備份之原始程式碼複製至新電腦，或下載已上傳至評分主機的程式碼。
- 七、對考題有任何疑義，請以評分系統之提問功能 (Communication) 提問 (請參考評分系統之使用手冊) 或填寫「提問單」交付監試人員轉送命題委員提出問題。監試人員不負責解答任何有關試題的問題。
- 八、競賽時間結束後，**參賽選手應留在位置上**等待評審前來共同確認評審結果。
- 九、試場守則：
 - 甲、試場內**不得與其他選手討論、上傳與解題無關檔案或惡意干擾線上自動評分系統**運作，承辦單位得依據違規狀況予以警告或取消比賽資格。
 - 乙、考試進行期間，選手可至試場外飲水、用餐、如廁，但**不得與他人研討試題**，違者得取消比賽資格。
 - 丙、離開試場，返回試場時，均須於試場入口填寫「離場休息登記表」。
 - 丁、離開試場時間，除午餐為 **1** 小時外，皆不可超過 **10** 分鐘。
 - 戊、考試時間用餐點、如廁均不可**離開工程三館三樓電梯北側區域**。
 - 己、午餐便當將於 11:30~12:30 三樓電梯門口旁供應。
 - 庚、午餐便當如未能及時食用，請於 14:30 前至三樓電梯門口領取，可使用微波爐加熱服務。
 - 辛、參賽中途如遇緊急事件(如停電等)，因應措施得依據命題及評審委員會決議處理。

(本頁空白)

一、機器人的位置

時間限制：3 秒

問題敘述：

研究人員開發了一套遙控系統，用它來控制一個在遙遠星球表面上的機器人移動。這套遙控系統的操作很簡單，它會傳送一連串指令給機器人，機器人接收指令後，會按照指令進行移動。你的工作是開發一個程式幫助研究人員，用其計算機器人移動後，最終的二維位置座標。

輸入說明：

測試資料每一行都會有一個指令。第一個是開始指令 B，之後兩個是 x 座標和 y 座標的設定，接下來是移動指令組，最後是結束指令 E。詳細的指令格式如下：

開始指令：B

設定座標指令：X <非負整數> 或 Y <非負整數>

移動指令(至少一組。最多 100 組)：

U <非負整數> 或 D <非負整數> 或 L <非負整數> 或 R <非負整數>

結束指令：E

指令字元與<非負整數>之間有一個或多個空白字元。<非負整數>之數值不超過 65,836。

指令字元的意義如下列表：

B：開始移動指令

X：設定機器人的 x 座標為<非負整數>

Y：設定機器人的 y 座標為<非負整數>

U：往正 y 軸方向移動<非負整數>格

D：往負 y 軸方向移動<非負整數>格

R：往正 x 軸方向移動<非負整數>格

L：往負 x 軸方向移動<非負整數>格

E：結束移動指令

輸出說明：

針對每個測試資料，第一行是 x 座標，第二行是 y 座標。

子題(Subtask)說明：

本題採 IOI 模式，在此題中有 5 個子題，每一個子題的時間限制都相同。最佳的解法，對每一個子題都可以在規定時限解出，如果你無法解決所有子題，也可以只解其中某些子題。你的成績將是你所繳交程式中分數最高者。

Subtask 1~5: 各 20%。

輸入範例 1：

B
X 100
Y 200
U 100
D 30
R 5
L 110
E

輸出範例 1：

-5
270

輸入範例 2：

B
X 10
Y 15
U 1
L 11
E

輸出範例 2：

-1
16

二、紙貼藝術

時間限制：3 秒

問題敘述：

快樂幼稚園的幼兒們正在學習藝術創作。幼兒們練習剪出半徑不一的半圓形紙條如下圖所示。剪完紙條後，再把這些半圓形紙條在端點處一對一的連接起來成為紙貼藝術作品，作品的每一紙條的每個端點都必須在同一平面上連接到另一紙條的兩個端點之一，但是紙條本身不可以變形(需維持原來的半圓形狀)因此該藝術品是由多個半圓紙條串接而成的平面作品(如圖一、二所示)。給定 n 個半圓形紙條的半徑，請找出最多可以用多少條紙條做出一個紙貼藝術作品？



圖一、紙條半徑長度為 2, 4, 2



圖二、紙條半徑長度為 2, 3, 3, 6

技術限制：

1. 紙條數量 n , 且 $2 \leq n \leq 10,000$
2. 紙條半徑 r_1, r_2, \dots, r_n , 且 $1 \leq r_1, r_2, \dots, r_n \leq 10,000$ 。
3. 為了方便計算，紙條寬度可以假設為無限小(圓心到紙條內圈或外圈距離皆為 r_i)

輸入說明：

每一筆測試資料都有兩行數字，第一行有一整數 n ，代表紙條數量，若 $n=0$ ，則結束測試。第二行有 n 個以空白隔開之正整數，分別代表 n 個紙條的半徑 r_1, r_2, \dots, r_n 。

輸出說明：

每一筆測試資料輸出一整數代表最多可用幾條紙條做出紙貼藝術作品。若無法做出紙貼藝術品，則輸出 0。

子題(Subtask)說明：

本題採 IOI 模式，在此題中有 5 個子題，每一個子題的時間限制都相同。最佳的解法，對每一個子題都可以在規定時限解出，如果你無法解決所有子題，也可以只解其中某些子題。你的成績將是你所繳交程式中分數最高者。

Subtask 1: (10%) 測試資料至多 6 筆， $2 \leq n \leq 10$ ， $1 \leq r_1, r_2, \dots, r_n \leq 10$ 。

Subtask 2: (20%) 測試資料至多 6 筆， $300 \leq n \leq 500$ ， $1 \leq r_1, r_2, \dots, r_n \leq 1,000$ 。

Subtask 3: (20%) 測試資料至多 6 筆， $300 \leq n \leq 500$ ， $1 \leq r_1, r_2, \dots, r_n \leq 5,000$ 。

Subtask 4: (20%) 測試資料至多 6 筆， $500 \leq n \leq 10,000$ ， $1 \leq r_1, r_2, \dots, r_n \leq 1,000$ 。

Subtask 5: (30%) 測試資料至多 6 筆， $2 \leq n \leq 10,000$ ， $1 \leq r_1, r_2, \dots, r_n \leq 10,000$ 。

輸入範例 1：

```
3
2 4 2
4
2 3 3 6
0
```

輸出範例 1：

```
3
4
```

輸入範例 2：

```
3
1 10 100
7
50 200 40 30 20 10 1200
0
```

輸出範例 2：

```
0
5
```

範例說明：

輸入範例 1 的兩組測試資料分別對應至圖一與圖二的例子。

三、分組尋寶

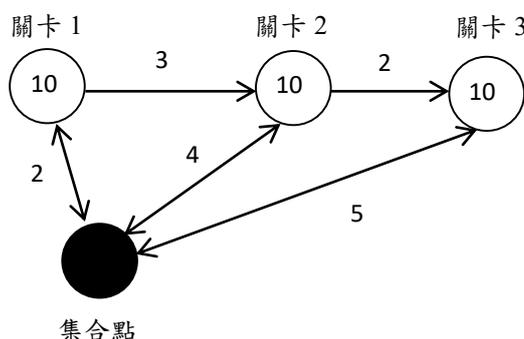
時間限制：3 秒

問題敘述：

身為指揮官，你現在帶領隊伍進入一個虛擬冒險遊戲中，所有人員正在集合點待命。你的任務是派員儘可能搜集寶藏，使得搜集的寶藏總價值越高越好。在搜集完寶藏後，所有人會從集合點離開這個冒險遊戲而回到現實世界來。遊戲中有 N 個關卡。關卡之間的傳送只有兩種方式：

(1) 關卡 i 可以傳送至關卡 $(i+1)$ ， $i = 1, \dots, N-1$ 。傳送耗費能量為 c_i 。

(2) 集合點可以傳送至任何關卡 i ，任何關卡 i 也都可以傳送回集合點，傳送一趟耗費的能量為 d_i 。如果傳送到關卡 i 後立即回集合點，去回總計需要 $2 \cdot d_i$ 的能量。



關卡 i 的寶藏價值為 p_i 。每個人的能量包可儲存的能量上限為 B ，必須在能量花費超過 B 之前回到集合點來，否則隊員便會陣亡，其搜集的寶藏也不予計算。隊員一旦返回集合點之後，就必須回到現實世界，不能再次前往關卡取得寶藏。此外，每個關卡的寶藏被拿走之後就不會再生，所以同一個關卡即使派人前往兩次，也只會得到價值 p_i 的寶藏，而非 $2 \cdot p_i$ 。指揮官，請下令派員出發搜集寶藏吧！

輸入說明：

第一行有一個整數 T ($T \leq 20$)，表示測試資料組數。

每一組測試資料的第一行有一個整數 N ($1 \leq N \leq 2,000$)，表示關卡數目。接下去有 N 行，每行有三個非負整數，代表關卡 $1, 2, \dots, N$ 的寶藏價值 p_i 、與集合點之間的傳送能量 d_i 以及到下一個關卡的傳送能量 c_i 。整數之間以一個空白隔開。最後一個關卡的 d_i 和 c_i 相等。此三整數值均小於 2,000。最後一行有兩個數字，以一個空白隔開，分別代表能量包的能量上限 B ($B \leq 1,000,000$) 和可派遣的人員數上限 R ($1 \leq R \leq N$)。

輸出說明：

針對每個測試資料，輸出能獲取寶藏的最大價值。

子題(Subtask)說明：

本題採 IOI 模式，在此題中有 5 個子題，每一個子題的時間限制都相同。最佳的解法，對每一個子題都可以在規定時限解出，如果你無法解決所有子題，也可以只解其中某些子題。你的成績將是你所繳交程式中分數最高者。

Subtask 1: (10%) $N \leq 5$

Subtask 2: (10%) $N \leq 20$

Subtask 3: (10%) $N \leq 40$

Subtask 4: (30%) $N \leq 100$

Subtask 5: (40%) $N \leq 2,000$

輸入範例 1：

```
2
3
10 2 3
10 4 2
10 5 5
11 1
3
10 2 3
10 4 2
10 5 5
11 2
```

輸出範例 1：

```
20
30
```

輸入範例 2：

```
2
1
100 10 10
20 1
1
100 10 10
10 1
```

輸出範例 2：

```
100
0
```

四、高速鐵路系統

時間限制：10 秒

問題敘述：

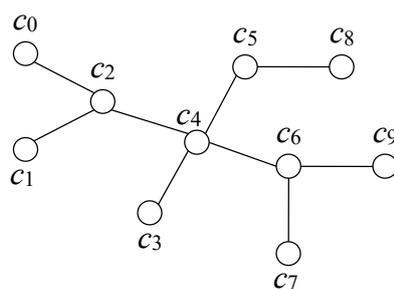
泰瓦德是一個美麗的國家。在泰瓦德中有 n 個都市，並且有 $n-1$ 條高速公路將它們連接起來。每一條高速公路都是雙向的，連接兩個不同的都市。此外，任兩個都市之間都只有唯一的一條由高速公路構成的路徑相連接。

每年有成千上萬的旅客來到泰瓦德的都市觀光。泰瓦德的首長計畫在其中一個都市打造一座主題樂園——水世界，以吸引更多旅客前來。為方便旅客前往水世界觀光，首長也計畫在水世界附近建造一套超高速鐵路系統，他的想法如下：

- (1) 選擇一些都市作為入口都市。這些入口都市必須滿足以下條件：如果都市 A 是入口都市，那麼從 A 到水世界所經過的所有都市，包含水世界本身，也是入口都市。
- (2) 在每一個入口都市，旅客可免費搭乘超高速列車前往水世界。

超高速鐵路系統的大小為所有入口都市之間的高速公路長度總和。由於預算有限，超高速鐵路系統的大小不能超過給定的長度 L 。

在泰瓦德的旅客可以先從他目前所在的都市前往一個最接近的入口都市，接著再搭乘免費的超高速列車，就能快速抵達水世界。從一個都市 X 到超高速鐵路系統的距離就是從 X 走到一個最近的入口都市的路徑總長度。首長希望所有都市到超高速鐵路系統的距離中，最遠的距離要越小越好。你的工作就是去幫助首長計算出如何建造最好的超高速鐵路系統。



圖一

假設 $L=40$ 。以圖一為例，其中 $n=10$ ，點代表都市，邊代表都市之間的高速公路。在這個例子中，所有高速公路長度都是 10。若選擇水世界位於 C_2 ，而 C_0, C_2, C_3, C_4, C_6 是入口都市，則超高速鐵路系統的大小是 40，離超高速鐵路系統的最遠的都市是 C_8 ，距離是 20 (從 C_8 到 C_4)。如果選擇 C_2, C_4, C_5, C_6 為入口都市，水世界位於 C_2 (或是任一個入口城市)，那麼這個超高速鐵路系統的大小是 30，離超高速鐵路系統最遠的距離是 10。在這個例子中，第二種選擇方式是能夠讓離超高速鐵路系統最遠的距離最小的最佳選擇方式之一。

輸入說明：

第一行有一個整數 T ($T \leq 5$)，表示接下來有 T 筆測試資料。每筆測試資料的第一行有兩個整數 n 及 L ，其中 $1 \leq n \leq 10^6$ ， $0 \leq L \leq 2 \times 10^8$ 。表示有 n 個都市 $0, 1, 2, \dots, n-1$ ，以及所建造的超高速鐵路系統的大小不能超過 L 。接下來有 $n-1$ 行，每一行有三個整數 i, j, d ，代表都市 i 和 j 之間有高速公路，長度為 d 。所有的高速公路長度 d 介於 1 與 2,000 之間。

輸出說明：

每筆測試資料必須輸出一個整數，表示在最好的建造方式下，離超高速鐵路系統最遠的都市的距離。

子題(Subtask)說明：

本題採 IOI 模式，在此題中有 5 個子題，每一個子題的時間限制都相同。最佳的解法，對每一個子題都可以在規定時限解出，如果你無法解決所有子題，也可以只解其中某些子題。你的成績將是你所繳交程式中分數最高者。

Subtask 1: (11%) $1 \leq n \leq 10,000$ ， $L=0$ 。而且這 $n-1$ 條高速公路會形成一條路徑 (path)，也就是說，每個城市最多連接兩條高速公路。

Subtask 2: (13%) $1 \leq n \leq 10,000$ ， $L \leq 2 \times 10^8$ 。而且這 $n-1$ 條高速公路會形成一條路徑 (path)。

Subtask 3: (21%) $1 \leq n \leq 10,000$ ， $L=0$ 。

Subtask 4: (25%) $1 \leq n \leq 10,000$ ， $L \leq 2 \times 10^8$ 。

Subtask 5: (30%) $1 \leq n \leq 10^6$ ， $L \leq 2 \times 10^8$ 。

輸入範例 1：(subtask 1)

```
2
3 0
0 1 30
1 2 70
8 0
2 5 6
1 2 5
5 6 6
3 1 9
6 7 15
7 0 12
4 3 12
```

輸出範例 1：

```
70
33
```

輸入範例 2 : (subtask 2)

1
8 33
2 5 6
1 2 5
5 6 6
3 1 9
6 7 15
7 0 12
4 3 12

輸出範例 2 :

21

輸入範例 3 : (subtask 3)

1
10 0
0 2 10
1 2 10
2 4 10
3 4 10
4 5 10
4 6 10
5 8 10
6 7 10
6 9 10

輸出範例 3 :

20

輸入範例 4 : (subtask 4~5)

1
10 40
0 2 10
1 2 10
2 4 10
3 4 10
4 5 10
4 6 10
5 8 10
6 7 10
6 9 10

輸出範例 4 :

10

注意事項：

使用 C++ 作答的同學，請在程式碼開頭加上 `#include<cstdio>`，並利用 `scanf` 讀入資料。使用 `cin` 讀入資料可能會因為讀入效率太差以致於程式執行時間超過限制。`scanf` 常用的讀入方式如下：

`scanf("%d", &x);` 讀入一個有號整數至 `int` 型態變數 `x`。

`scanf("%lld", &y);` 讀入一個有號整數至 `long long` 型態變數 `y`。

`scanf("%u", &x);` 讀入一個無號整數至 `unsigned int` 型態變數 `x`。

`scanf("%llu", &y);` 讀入一個無號整數至 `unsigned long long` 型態變數 `y`。

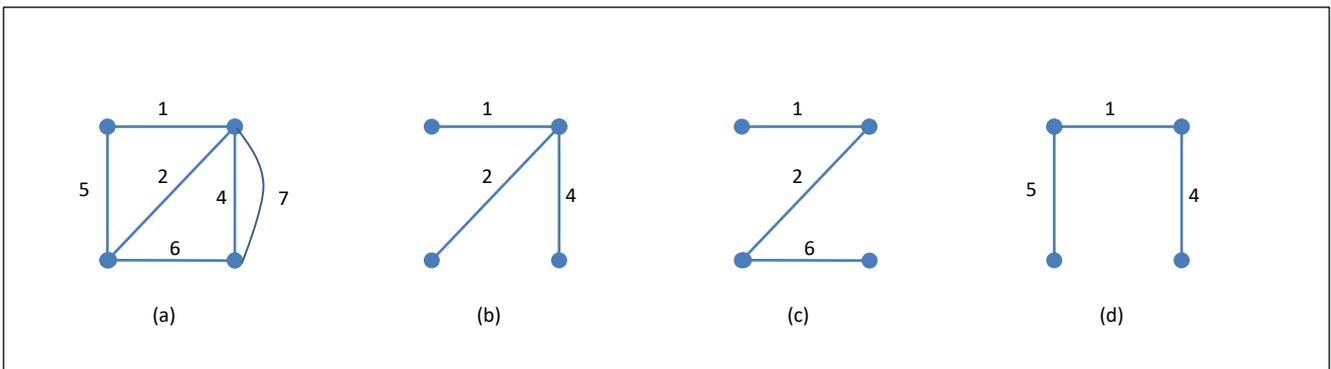
五、最小與次小網路建構成本

時間限制：3 秒

問題敘述：

某連鎖商店有 N 個分店，總公司打算建構一個網路將所有據點的電腦系統連接起來。經過顧問公司所做的調查，某些分店之間可以架設線路，但有些不能。架設線路也各自需要不同的成本。該公司現在對外招標連線的設計，由於該公司還有其他的考量，所以希望每一個投標廠商必須提出兩個不同的網路連線設計。根據該公司公告的可能線路成本資料，你的工作是計算出最佳以及次佳網路的建構成本之差值。

N 個分店的編號為 $0 \sim N-1$ ，要將這些分店連接成一個網路，至少需要 $N-1$ 條連線。把每一個分店看成一個點，點與點之間有一些具某些成本的邊，邊代表可以選擇的連線成本，一個網路設計就是要選出某些邊將所有的點連成一個連通的區塊使得任兩點之間皆可直接或間接相連，而其成本就是所選擇的邊的成本總和，在這個問題中，你要求的是**最小成本**和**次小成本**的網路設計。所謂次小成本，是指成本大於最小成本的解中，成本最小的。



圖一

以圖一為例，(a)是輸入資料，包含四個點和六個可能的連線；(b)是將四個點連成連通網路的最小成本，其成本為 $1+2+4=7$ ；(c)是次小成本，其成本為 9；(d)的成本是 10，因此他不是本題所求的答案。請留意，如圖一(a)，本題中兩點之間可能有超過一個邊，次小成本有可能使用這些邊。在此例中，最小與次小的成本差值就是 $9-7=2$ 。

輸入說明：

第一行有一個整數 T ($T \leq 8$)，表示有 T 個測試資料。

每個測試資料的第一行有兩正整數 N 和 M ，代表共有 N 個分店需要被連接以及有 M 個可能的連線架設方式，分店的編號是 $0 \sim N-1$ 。接下來的 M 行每行有三個整數字表示一個可能的連線架設方式，前兩個數字是連接的兩個分店，第三個數字是這個連線的成本：一個不超過 500,000 的正整數。連線的成本皆不相同，所求答案不會超過 2^{31} 。

輸出說明：

針對每個測試資料，依序輸出最小與次小成本之差值，每個測試資料輸出一行。本題的所有測試案例都必然有最小與次小成本。

子題(Subtask)說明：

本題採 IOI 模式，在此題中有 5 個子題，每一個子題的時間限制都相同。最佳的解法，對每一個子題都可以在規定時限解出，如果你無法解決所有子題，也可以只解其中某些子題。你的成績將是你所繳交程式中分數最高者。

Subtask 1: (11%) $N \leq 100$, $M \leq 500$ 。且本子題的測試資料滿足以下性質：對所有 $i > 0$ ，分店 i 都和分店 0 有一個連線架設方式，其成本必定是分店 i 所有連線方式中成本最小的。

Subtask 2: (13%) $N \leq 100$, $M \leq 500$ 。對所有 $i > 0$ ，分店 i 都和分店 $i-1$ 有一個連線架設方式，其成本必定是分店 i 所有連線方式中成本最小的。

Subtask 3: (21%) $N \leq 100$, $M \leq 500$ 。

Subtask 4: (25%) $N \leq 2,000$, $M \leq 15,000$ 。

Subtask 5: (30%) $N \leq 20,000$, $M \leq 130,000$ 。

輸入範例 1：(subtask 1)

```
1
4 6
0 1 1
2 0 2
3 1 8
3 0 7
1 2 6
2 3 10
```

輸出範例 1：

```
1
```

輸入範例 2：(subtask 2)

```
2
3 4
0 1 3
2 1 4
0 2 6
1 2 5
5 5
0 1 1
1 2 3
3 2 4
3 4 8
0 4 10
```

輸出範例 2：

```
1
2
```

輸入範例 3 : (subtask 3~5)

```
1
6 7
0 1 1
0 2 2
1 3 3
1 4 4
2 5 5
5 3 6
1 3 7
```

輸出範例 3 :

```
1
```

注意事項 :

使用 C++ 作答的同學，請在程式碼開頭加上 `#include<cstdio>`，並利用 `scanf` 讀入資料。使用 `cin` 讀入資料可能會因為讀入效率太差以致於程式執行時間超過限制。`scanf` 常用的讀入方式如下：

`scanf("%d", &x);` 讀入一個有號整數至 `int` 型態變數 `x`。

`scanf("%lld", &y);` 讀入一個有號整數至 `long long` 型態變數 `y`。

`scanf("%u", &x);` 讀入一個無號整數至 `unsigned int` 型態變數 `x`。

`scanf("%llu", &y);` 讀入一個無號整數至 `unsigned long long` 型態變數 `y`。

(本頁空白)

六、石油生產問題

時間限制：3 秒

問題敘述：

中東 D 國的某石油公司有 M 座油田(以 $1, 2, \dots, M$ 表示)和 N 座煉油廠(以 $1, 2, \dots, N$ 表示)，油田與煉油廠間有 K 條生產線連結，每一個油田都至少有連結到一個煉油廠。油田 i 連結煉油廠 j 的生產線 (i, j) 開工提煉石油可獲得利潤 $w_{i,j}$ (以有理數表示： $w_{i,j} = \frac{q_{i,j}}{p_{i,j}}$)

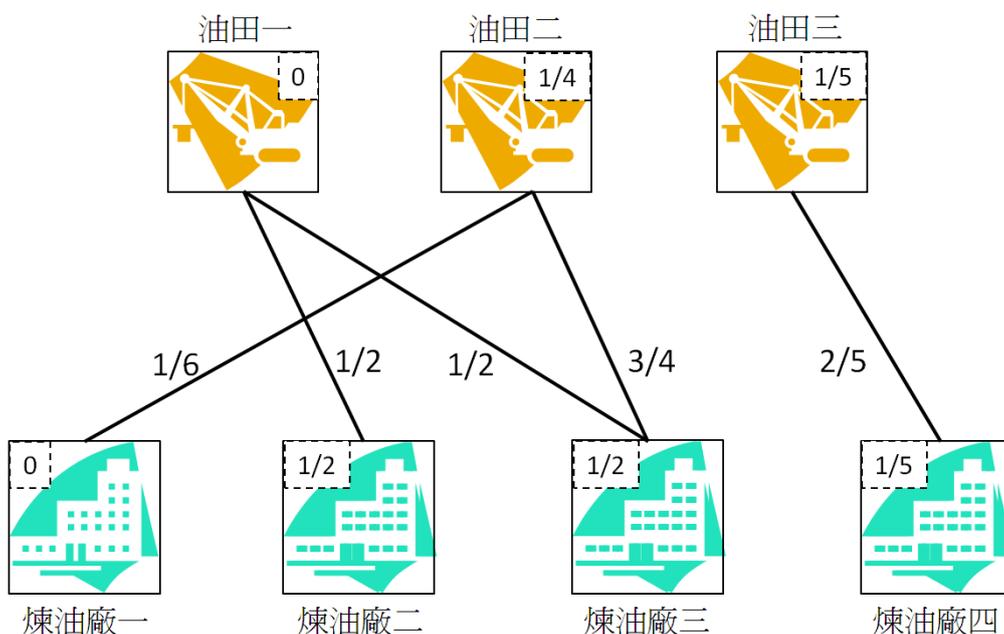
其中 $p_{i,j}$ 和 $q_{i,j}$ 為非負整數且 $p_{i,j} \neq 0$)，但共用油田或煉油廠的生產線不能同時開工。D 國政府宣稱現階段石油產量過多，導致油價下跌。為了止跌，D 國政府決定付出一些費用要求石油公司暫時停產，其策略如下：

- 假如石油公司不開採油田 i ，則政府願付出 u_i (u_i 為有理數) 給石油公司。
- 假如石油公司不使用煉油廠 j ，則政府願付出 v_j (v_j 為有理數) 給石油公司。

很顯然地，如果 $u_i + v_j < w_{i,j}$ ，則生產線 (i, j) 開工提煉石油，所獲得的利潤將比政府給的多。為了保證生產線 (i, j) 停工，D 國政府願意付出的金額，必需要滿足 $u_i + v_j \geq w_{i,j}$ 。所有的生產線都必須停工，才能讓石油公司暫時停產。

給定 M 座油田、 N 個煉油廠、 K 條生產線 (i, j) 及其利潤 $w_{i,j}$ ，請寫一個程式幫 D 國政府計算要讓所有生產線停產所需支付最少值(亦即最小化 $\sum_i u_i + \sum_j v_j$) 為何？

範例說明：假設有 3 座油田和 4 座煉油廠，並有 5 條生產線： $(1,2)$ 、 $(1,3)$ 、 $(2,1)$ 、 $(2,3)$ 、 $(3,4)$ ，可提供利潤 $w_{1,2} = \frac{1}{2}$ ； $w_{1,3} = \frac{1}{2}$ ； $w_{2,1} = \frac{1}{6}$ ； $w_{2,3} = \frac{3}{4}$ ； $w_{3,4} = \frac{2}{5}$ (如下圖所示)。則政府最少需支付 $u_1 = 0$ ； $u_2 = \frac{1}{4}$ ； $u_3 = \frac{1}{5}$ ； $v_1 = 0$ ； $v_2 = \frac{1}{2}$ ； $v_3 = \frac{1}{2}$ ； $v_4 = \frac{1}{5}$ ，亦即需付出的最小值為 $\sum_{i=1}^3 u_i + \sum_{j=1}^4 v_j = 0 + \frac{1}{4} + \frac{1}{5} + 0 + \frac{1}{2} + \frac{1}{2} + \frac{1}{5} = \frac{33}{20}$ 。



輸入說明：

第一行有一個數字 T ($T \leq 5$) 代表此子題測試資料的數目。每組測試資料的第一行有三個數字，代表 M 值、 N 值與 K 值，數字以空白隔開。接下來 K 行代表 K 條生產線，每條生產線有四個數字：第一個數字代表油田編號 i 、第二個數字代表煉油廠編號 j 、第三個數字代表此生產線的利潤值的分子 q_{ij} 、第四個數字代表此生產線的利潤值的分母 p_{ij} 。

輸出說明：

針對所輸入的資料，輸出一個有理數(最簡分數)代表政府所需支付石油公司的最小值。每行輸出兩個數字：第一個數字代表分子、第二個數字代表分母。

子題(Subtask)說明：

本題採 IOI 模式，在此題中有 5 個子題，每一個子題的時間限制都相同。最佳的解法，對每一個子題都可以在規定時限解出，如果你無法解決所有子題，也可以只解其中某些子題。你的成績將是你所繳交程式中分數最高者。

Subtask 1: (11%) $1 \leq M \leq 10$ 、 $1 \leq N \leq 10$ 、 $M \leq K \leq MN$ 、 $p_{ij}=1$ 、 $1 \leq q_{ij} \leq 10$ 。

Subtask 2: (13%) $1 \leq M \leq 100$ 、 $1 \leq N \leq 100$ 、 $M \leq K \leq MN$ 、 $p_{ij}=1$ 、 $q_{ij}=1$ 。

Subtask 3: (21%) $1 \leq M \leq 100$ 、 $1 \leq N \leq 100$ 、 $M \leq K \leq MN$ 、 $p_{ij}=1$ 、 $1 \leq q_{ij} \leq 10^6$ 。

Subtask 4: (25%) $1 \leq M \leq 500$ 、 $1 \leq N \leq 500$ 、 $M \leq K \leq MN$ 、 $1 \leq p_{ij} \leq 10$ 、 $1 \leq q_{ij} \leq 5,000$ 。

Subtask 5: (30%) $1 \leq M \leq 500$ 、 $1 \leq N \leq 500$ 、 $M \leq K \leq MN$ 、 $1 \leq p_{ij} \leq 10$ 、 $1 \leq q_{ij} \leq 10^6$ 。

輸入範例 1：(subtask 1)

```
1
2 2 4
1 1 1 1
1 2 2 1
2 1 2 1
2 2 4 1
```

輸出範例 1：

```
5 1
```

輸入範例 2：(subtask 2)

```
1
2 2 4
1 1 1 1
1 2 1 1
2 1 1 1
2 2 1 1
```

輸出範例 2：

```
2 1
```

輸入範例 3 : (subtask 3~5)

```
1
3 4 5
1 2 1 2
1 3 1 2
2 1 1 6
2 3 3 4
3 4 2 5
```

輸出範例 3 :

```
33 20
```

注意事項 :

使用 C++作答的同學，請在程式碼開頭加上 `#include<cstdio>`，並利用 `scanf` 讀入資料。使用 `cin` 讀入資料可能會因為讀入效率太差以致於程式執行時間超過限制。`scanf` 常用的讀入方式如下：

`scanf("%d",&x);` 讀入一個有號整數至 `int` 型態變數 `x`。

`scanf("%lld",&y);` 讀入一個有號整數至 `long long` 型態變數 `y`。

`scanf("%u",&x);` 讀入一個無號整數至 `unsigned int` 型態變數 `x`。

`scanf("%llu",&y);` 讀入一個無號整數至 `unsigned long long` 型態變數 `y`。