

## دستگیر کردن دزدها

- محدودیت زمان: ۱۰ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

در این مسئله،  $N$  پلیس وجود دارد که با 1 تا  $N$  شماره‌گذاری شده‌اند و  $M$  دزد وجود دارد که با 1 تا  $M$  شماره‌گذاری شده‌اند. تمامی افراد (پلیس‌ها و دزدان) نقطه‌ای دوبعدی در یک صفحه دکارتی هستند. مختصات پلیس  $i$ -ام را با  $(Xp_i, Yp_i)$  و مختصات دزد  $i$ -ام را با  $(Xt_i, Yt_i)$  نشان می‌دهیم.

یک دزد دستگیر می‌شود اگر مجموعه‌ای از پلیس‌ها وجود داشته باشد که یک چندضلعی محاط‌کننده بسازند به طوری که دزد اکیداً داخل (و نه روی) آن چندضلعی قرار بگیرد.

اداره پلیس می‌خواهد تعدادی (صفر یا بیشتر) پلیس به عنوان نیروی تقویتی اعزام کند تا مطمئن شود که تمامی دزدان دستگیر می‌شوند. مختصات این پلیس‌های اضافی به دلخواه انتخاب می‌شود (می‌توانند هر مقدار حقیقی‌ای داشته باشند)؛ اما اجازه‌ی جابه‌جایی پلیس‌هایی که در ابتدا حضور داشتند وجود ندارد. کمترین تعداد پلیس‌هایی که باید به عنوان نیروی تقویتی برای دستگیری تمامی دزدان اعزام شوند را محاسبه کنید.

## ورودی

- اولین خط ورودی شامل یک عدد صحیح  $T$  است که تعداد سناریوها را نشان می‌دهد. سپس توصیف  $T$  سناریو در ادامه می‌آید.
- اولین خط هر مورد سناریو شامل دو عدد صحیح جدا شده با فاصله  $N$  و  $M$  است (اول  $N$  می‌آید و سپس  $M$ ).
- $N$  خط بعدی دنبال می‌شود. برای هر  $i$  ( $1 \leq i \leq N$ )، خط  $i$ -ام از این خطوط دو عدد صحیح جدا شده با فاصله  $Xp_i$  و  $Yp_i$  را در خود دارد.
- $M$  خط دیگر دنبال می‌شود. برای هر  $i$  ( $1 \leq i \leq M$ )، خط  $i$ -ام از این خطوط دو عدد صحیح جدا شده با فاصله  $Xt_i$  و  $Yt_i$  را در خود دارد.

## خروجی

برای هر سناریو، یک خط شامل یک عدد صحیح چاپ کنید: حداقل تعداد افسران پلیس اضافی. (پاسخ می‌تواند صفر هم باشد).

## محدودیت‌ها

- $1 \leq T \leq 10$
- $0 \leq N \leq 10^5$
- $1 \leq M \leq 10^5$
- برای هر  $i$  معتبر  $|Xp_i|, |Yp_i| \leq 2 \times 10^8$
- برای هر  $i$  معتبر  $|Xt_i|, |Yt_i| \leq 2 \times 10^8$
- هیچ دو نفر (افسران پلیس یا دزدها) در یک موقعیت یکسان نیستند.

## مثال

### ورودی نمونه ۱

```
1
1 1
10 10
20 20
```

### خروجی نمونه ۱

```
2
```

## حساب و کتاب

- محدودیت زمان: ۱۰ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

افرادی را می‌شناسیم که می‌دانیم به مرور زمان به یکدیگر پول قرض می‌دهند. در ابتدا هیچ‌کس به دیگری بدهکار یا طلبکار نیست اما در طی این قرض دادن‌ها افراد نسبت به یکدیگر بدهکار یا طلبکار می‌شوند. در این مسئله باید با بررسی گزارش مالی این افراد، مقداری که نسبت به هم بدهکار یا طلبکار می‌شوند را محاسبه کنید.

بدهکار و طلبکار بودن یا نبودن افراد را طبق همان تعاریف بدیهی‌ای که در واقعیت داریم چنین تعریف می‌کنیم:

- فرد  $x$  به فرد  $y$  بدهکار است، اگر و تنها اگر مقداری اکیداً بزرگ‌تر از 0.00 دلار وجود داشته‌باشد که زمانی قبل‌تر  $y$  به  $x$  قرض داده‌باشد اما هنوز آن را کامل پس نگرفته‌باشد.
- فرد  $x$  از فرد  $y$  طلبکار است، اگر و تنها اگر مقداری اکیداً بزرگ‌تر از 0.00 دلار وجود داشته‌باشد که زمانی قبل‌تر  $x$  به  $y$  قرض داده‌باشد اما هنوز آن را کامل پس نگرفته‌باشد.

گزارش مالی در طی زمان به صورت تعدادی دستور برای شما ارسال می‌شود و شما هر بار باید این دستورها را روی ساختمان داده‌ی خود اعمال کنید و اگر نیاز بود اطلاعاتی را هم چاپ کنید.

دستورهایی که به شما داده می‌شود، در قالب یکی از چند حالت زیر است:

1.  $s_1 \ s_2 \ x$  : این دستور، یعنی فردی با نام  $s_1$  مقدار  $x$  دلار به فردی با نام  $s_2$  قرض داد.
2.  $s_1 \ s_2$  : با این دستور، از شما درخواست می‌شود که نام کسی را چاپ کنید که مجموع میزان سرمایه‌ی به‌دست آورده‌اش منهای مجموع میزان سرمایه‌ی از دست داده‌اش **بیشینه (و مثبت)** باشد. اگر چندین فرد این ویژگی را داشتند، نام فردی را بنویسید که از نظر ترتیب لغتنامه‌ای کوچک‌تر است. اگر هیچ فردی که مجموع میزان سرمایه‌ی به‌دست آورده‌اش از مجموع میزان سرمایه‌ی از دست داده‌اش اکیداً بیش‌تر باشد وجود نداشت، فقط 1- چاپ کنید.

۳. 3 : با این دستور، از شما درخواست می‌شود که نام کسی را چاپ کنید که مجموع میزان سرمایه‌ی به‌دست آورده‌اش منهای مجموع میزان سرمایه‌ی از دست داده‌اش کمینه (و منفی) باشد. اگر چندین فرد این ویژگی را داشتند، نام فردی را بنویسید که از نظر ترتیب لغت‌نامه‌ای کوچک‌تر است. اگر هیچ فردی که مجموع میزان سرمایه‌ی به‌دست آورده‌اش از مجموع میزان سرمایه‌ی از دست داده‌اش اکیداً کمتر باشد وجود نداشت، فقط 1- چاپ کنید.

۴. 4 s : با این دستور، از شما درخواست می‌شود که تعداد افرادی را چاپ کنید که فردی با نام s به آن‌ها بدهکار است.

۵. 5 s : با این دستور، از شما درخواست می‌شود که تعداد افرادی را چاپ کنید که فردی با نام s از آن‌ها طلبکار است.

۶. 6 s\_1 s\_2 : با این دستور، از شما درخواست می‌شود که چاپ کنید فردی با نام s\_1 دقیقاً چند دلار باید به فردی با نام s\_2 بدهد تا حسابشان صاف شود. این مقدار می‌تواند مثبت یا صفر یا منفی باشد و باید دقیقاً با دو رقم اعشار چاپ شود (مقدار منفی به‌جای دادن پول، گرفتن پول را نشان می‌دهد).

## ورودی

در اولین خط ورودی عدد صحیح q آمده‌است که تعداد دستورها را نشان می‌دهد.

$$1 \leq q \leq 1.5 \times 10^5$$

در q خط بعدی، هر خط از ورودی یکی از شش حالتی که در بالا بیان شد را دارد. تمام اسامی افراد فقط شامل حروف کوچک الفبای انگلیسی می‌شوند و تعداد حروف هیچ‌یک از ۸ بیش‌تر نیست. در همه‌ی انواع دستورها به‌جز دستوره‌ای نوع a، تضمین می‌شود که نام فردی که در ورودی می‌آید جدید نیست (یعنی نامش پیش از آن حداقل یک بار دیگر هم در ورودی آمده‌است). تمام مقادیر عددی مالی (که به واحد دلار نوشته شده‌اند)، دقیقاً دارای ۲ رقم اعشار می‌باشند. این اعداد مثبت هستند و کمتر از 0.01 نیستند و بیش‌تر از 10000000.99 هم نیستند.

## خروجی

به ازای دستوره‌ای نوع ۲، ۳، ۴، ۵ و ۶ پاسخ درخواست را چاپ کنید (و به ازای دستوره‌ای نوع ۱، در هیچ حالتی چیزی چاپ نکنید).

## مثال

### ورودی نمونه ۱

```
21
1 mohsen hamid 5.50
1 hamid mohsen 5.50
1 ali mohsen 15.50
1 mohsen ali 15.50
1 ali reza 10.00
6 reza ali
1 reza ali 30.00
1 ali reza 40.00
6 reza ali
2
3
4 ali
5 ali
6 ali reza
1 reza ali 21.00
2
3
1 ali reza 1.00
2
3
6 ali reza
```

### خروجی نمونه ۱

```
10.00
20.00
reza
```

ali  
0  
1  
-20.00  
ali  
reza  
-1  
-1  
0.00

ورودی نمونه ۲

27  
1 a b 100.00  
2  
3  
1 b c 100.00  
2  
3  
6 a b  
6 b c  
6 c a  
6 b a  
6 c b  
6 a c  
1 c a 100.00  
2  
3  
6 a b  
6 b c  
6 c a  
6 b a  
6 c b  
6 a c  
4 a  
4 b  
4 c  
5 a

5 b

5 c

## خروجی نمونه ۲

b

a

c

a

-100.00

-100.00

0.00

100.00

100.00

0.00

-1

-1

-100.00

-100.00

-100.00

100.00

100.00

100.00

1

1

1

1

1

1

## نکات

- به ازای هر ورودی معتبر مسئله، دقیقاً یک خروجی مشخص درست وجود دارد.
- علامت منفی را فقط پشت اعداد منفی بگذارید. برای مثال چاپ کردن  $-0.00$  اشتباه است.

## درخت‌سازی

- محدودیت زمان: ۱۰ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

یک گراف ساده (گرافی بدون جهت و بدون وزن و بدون یال چندگانه و بدون طوقه) به شما داده می‌شود. شما باید کم‌ترین تعداد یال ممکن را از گراف حذف کنید و کم‌ترین تعداد یال ممکن را به گراف اضافه کنید، تا این گراف تبدیل به یک درخت (گراف هم‌بند بدون دور) بشود.

رئوس گراف با اعداد طبیعی از 1 تا  $n$  شماره‌گذاری شده‌اند.

## ورودی

در اولین خط ورودی دو عدد  $n$  و  $m$  با یک فاصله بینشان آمده‌است که به ترتیب تعداد رئوس و یال‌های گراف را نشان می‌دهد.

$$1 \leq n \leq 2000$$

$$0 \leq m \leq \frac{n(n-1)}{2}$$

در ادامه  $m$  خط در ورودی آمده‌است. در  $i$ امین خط از  $m$  خط بعدی، دو عدد طبیعی  $x_i$  و  $y_i$  نابرابر با یک فاصله بینشان آمده‌است که نشان می‌دهد یال  $(x_i, y_i)$  در گراف وجود دارد.

به ازای هر  $i$  معتبر داریم:

$$1 \leq x_i, y_i \leq n$$

$$x_i \neq y_i$$



## خروجی

در اولین خط، اول  $a$  (تعداد یال‌های لازم برای حذف شدن از گراف) و سپس  $b$  (تعداد یال‌های لازم برای اضافه شدن به گراف) را با یک فاصله چاپ کنید. در هر یک از  $a$  خط بعدی، شماره‌ی رئوس دو سر هر یالی که لازم است از گراف حذف شود را با یک فاصله چاپ کنید. سپس در هر یک از  $b$  خط بعدی، شماره‌ی رئوس دو سر هر یالی که لازم است به گراف اضافه شود را با یک فاصله چاپ کنید.

## مثال

### ورودی نمونه ۱

3 2  
1 2  
2 3

### خروجی نمونه ۱

0 0

### ورودی نمونه ۲

3 3  
1 2  
2 3  
3 1

### خروجی نمونه ۲

1 0

1 3

ورودی نمونه ۳

4 0

خروجی نمونه ۳

0 3

1 2

2 3

3 4

نکات

- هر جواب دلخواهی که گراف ورودی را تبدیل به یک درخت بکند معتبر خواهد بود، ترتیب چاپ کردن یال‌های خروجی مهم نیست، فقط دقت کنید که مقدار  $a + b$  باید کمینه باشد، یعنی مجموع تعداد یال‌های حذف‌شده و یال‌های اضافه‌شده باید کمینه باشد.

## انتخاب رشته

- محدودیت زمان: ۱ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۲۵۶ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

به نظر عرفان، یک معیار مهم در انتخاب دانشگاه میزان نزدیکی آن دانشگاه به محل زندگی متقاضی است. عرفان معتقد است افرادی که در شرق تهران زندگی می‌کنند و قصد انتخاب رشته برای دانشگاه‌های سراسری تهران را دارند، بهتر است دانشگاه علم و صنعت را در اولویت قرار دهند. او همچنین معتقد است افرادی که در غرب تهران زندگی می‌کنند، بهتر است دانشگاه سراسری دیگری را در اولویت قرار دهند.

به افرادی که به تازگی در کنکور سراسری کارشناسی شرکت کرده‌اند کمک کنید تا ببینند اگر بخواهند طبق نظر عرفان عمل کنند، باید دانشگاه علم و صنعت را در اولویت قرار دهند یا خیر.

## ورودی

ورودی تنها شامل یک خط است که در آن عبارت EAST یا NOT EAST ظاهر شده است. اولی نشان‌دهنده سکونت متقاضی در شرق است و دومی نشان‌دهنده سکونت متقاضی در منطقه دیگری از تهران است. (تمامی حروف انگلیسی بزرگ هستند.)

## خروجی

اگر برای متقاضی بهتر است که دانشگاه علم و صنعت را در اولویت قرار دهد، YES چاپ کنید و در غیر این صورت NO چاپ کنید. (تمامی حروف انگلیسی بزرگ هستند.)

## مثال

### ورودی نمونه ۱

EAST

خروجی نمونه ۱

YES

ورودی نمونه ۲

NOT EAST

خروجی نمونه ۲

NO

## رشته‌های خفن

- محدودیت زمان: ۶ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

یک رشته با طول  $L$  را خفن می‌نامیم اگر  $L \geq 3$  باشد و یک کاراکتر وجود داشته باشد که اکیداً بیش از  $L/2$  بار در این رشته ظاهر شود.

رشته‌ای به نام  $S$  به شما داده شده است و شما باید  $Q$  پرسش را در مورد این رشته پاسخ دهید. در هر پرسش، یک زیررشته‌ی پیوسته  $S_L, S_{L+1}, \dots, S_R$  به شما داده می‌شود. تمام زیررشته‌های پیوسته‌ی این زیررشته را در نظر بگیرید. شما باید تعیین کنید که آیا حداقل یکی از آن‌ها خفن است یا خیر.

## ورودی

- اولین خط ورودی شامل یک عدد صحیح  $T$  است که تعداد سناریوها را نشان می‌دهد. سپس توضیحات  $T$  سناریو به دنبال آن می‌آید.
- اولین خط هر سناریو شامل دو عدد صحیح جداشده با فاصله،  $N$  و  $Q$  است (اول  $N$  می‌آید و سپس  $Q$ ).
- خط دوم شامل یک رشته‌ی  $S$  با طول  $N$  است.
- هر یک از  $Q$  خط بعدی شامل دو عدد صحیح جداشده با فاصله‌ی  $L$  و  $R$  است که یک پرسش را توصیف می‌کند.

## خروجی

برای هر پرسش، یک خط شامل YES چاپ کنید اگر زیررشته‌ی داده‌شده شامل حداقل یک زیررشته‌ی خفن باشد و یا NO اگر هیچ زیررشته‌ی خفنی در آن وجود نداشته باشد.

## محدودیت‌ها

- $1 \leq T \leq 10$
- $1 \leq N, Q \leq 10^5$
- $1 \leq L \leq R \leq N$
- رشته‌ی  $S$  فقط شامل حروف کوچک انگلیسی است.

## مثال

### ورودی نمونه ۱

```
1
10 2
helloworld
1 3
1 10
```

### خروجی نمونه ۱

```
NO
YES
```

## موش و گربه

- محدودیت زمان: ۱۰ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

در این مسئله،  $N$  گربه (با شماره‌های ۱ تا  $N$ ) و  $M$  موش (با شماره‌های ۱ تا  $M$ ) روی یک خط هستند. هر گربه و هر موش می‌خواهد از نقطه‌ای به سمت نقطه‌ای دیگر (شاید همان نقطه) روی این خط حرکت کند. طبیعتاً، گربه‌ها هم می‌خواهند موش‌ها را بخورند. هر گربه و هر موش با سرعت ثابت ۱ حرکت می‌کند.

برای هر  $i$  معتبر،  $i$ -امین گربه در ابتدا در نقطه‌ای  $a_i$  خواب است. در زمان  $s_i$ ، این گربه بیدار می‌شود و به سمت نقطه‌ی نهایی  $b_i$  با سرعت ثابت و بدون هیچ پرشی حرکت می‌کند (بنابراین در زمان  $e_i = s_i + |a_i - b_i|$  به این نقطه می‌رسد). پس از رسیدن به نقطه‌ی  $b_i$ ، دوباره به خواب می‌رود.

برای هر  $i$  معتبر،  $i$ -امین موش در ابتدا در نقطه‌ی  $c_i$  پنهان شده است. در زمان  $r_i$ ، این موش از پنهانی در می‌آید و به سمت نقطه‌ی نهایی  $d_i$  به همان صورتی که گربه‌ها حرکت می‌کنند - با سرعت ثابت و بدون هیچ پرشی - حرکت می‌کند و در زمان  $q_i = r_i + |c_i - d_i|$  (اگر خورده نشود) به این نقطه می‌رسد. پس از رسیدن به نقطه‌ی  $d_i$ ، دوباره پنهان می‌شود.

اگر یک گربه و یک موش یکدیگر را ببینند (یعنی در یک نقطه و در یک زمان قرار بگیرند)، گربه موش را می‌خورد و موش ناپدید می‌شود و نمی‌تواند توسط گربه‌ای دیگر خورده شود. یک گربه‌ی خوابیده نمی‌تواند موشی را بخورد و یک موش پنهان نمی‌تواند خورده شود - به صورت دقیق‌تر، گربه‌ی  $i$  تنها در صورتی می‌تواند موش  $j$  را بخورد که در زمان  $t$  با هم برخورد کنند به طوری که  $s_i \leq t \leq e_i$  و  $r_j \leq t \leq q_j$  برقرار باشد.

وظیفه‌ی شما این است که پیدا کنید کدام موش‌ها توسط کدام گربه‌ها خورده می‌شوند. تضمین می‌شود که دو گربه به طور همزمان با یک موش برخورد نمی‌کنند.

## ورودی

- خط اول ورودی شامل یک عدد صحیح  $T$  است که تعداد سناریوها را نشان می‌دهد. توضیحات مربوط به  $T$  سناریو در ادامه آمده است.
- خط اول هر سناریو شامل دو عدد صحیح  $N$  و  $M$  است که با یک فاصله از هم جدا شده‌اند (در ابتدا  $N$  می‌آید و سپس  $M$ ).
- خط  $N$  در ادامه آمده است. برای هر  $i$  ( $1 \leq i \leq N$ )، خط  $i$ -ام از این خطوط شامل سه عدد صحیح  $a_i, b_i, s_i$  است که با یک فاصله از هم جدا شده‌اند.
- خط دیگر در ادامه آمده است. برای هر  $i$  ( $1 \leq i \leq M$ )، خط  $i$ -ام از این خطوط شامل سه عدد صحیح  $c_i, d_i, r_i$  است که با یک فاصله از هم جدا شده‌اند.

## خروجی

برای هر سناریو،  $M$  خط چاپ کنید. برای هر  $i$  معتبر، خط  $i$ -ام از این خطوط باید شامل یک عدد صحیح باشد: شماره گربه‌ای که موش  $i$ -ام را می‌خورد و یا  $-1$  اگر هیچ گربه‌ای این موش را نخورده است.

## محدودیت‌ها

- $1 \leq T \leq 10$
- $0 \leq N \leq 1,000$
- $1 \leq M \leq 1,000$
- برای هر  $i$  معتبر  $1 \leq a_i, b_i, s_i \leq 10^9$
- برای هر  $i$  معتبر  $1 \leq c_i, d_i, r_i \leq 10^9$
- همه موقعیت‌های اولیه و نهایی همه‌ی گربه‌ها و موش‌ها متمایز هستند. البته درمورد هر موش خاص موقعیت شروع و پایان آن می‌تواند یکسان باشد. و درمورد هر گربه‌ی خاص موقعیت شروع و پایان آن می‌تواند یکسان باشد.

## مثال

### ورودی نمونه ۱



1  
8 8  
2 8 2  
12 18 2  
22 28 4  
32 38 4  
48 42 2  
58 52 3  
68 62 1  
78 72 3  
3 6 3  
13 19 3  
21 25 3  
31 39 3  
46 43 4  
59 53 2  
65 61 4  
79 71 2

خروجی نمونه ۱

1  
2  
3  
4  
5  
6  
7  
8

## خرید قانونی آهنگ

- محدودیت زمان: ۱۰ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

همانطور که همه می‌دانند، حسین نخبه هنرمند مورد علاقه‌ی ایرانیان است. از آن‌جا که ما حسین نخبه و آهنگ‌های او را دوست داریم، هرگز آن‌ها را به صورت رایگان دانلود نمی‌کنیم. باید آهنگ‌های حسین نخبه را از فروشگاه‌های مجاز خریداری کرد. اما اگر پول کافی برای خرید تمام آهنگ‌ها نباشد چه کار کنیم؟

برای ما  $N$  آهنگ (شماره‌گذاری شده از 1 تا  $N$ ) و  $M$  آلبوم (شماره‌گذاری شده از 1 تا  $M$ ) در دسترس وجود دارد. برای هر  $i$  معتبر، آهنگ  $i$ -ام ارزش موسیقایی  $v_i$  و  $p_i$  تومان قیمت دارد و به آلبوم  $a_i$  تعلق دارد. برای هر  $i$  معتبر، آلبوم  $i$ -ام  $b_i$  تومان قیمت دارد. اگر یک آلبوم را بخریم، همه‌ی آهنگ‌های این آلبوم به خریداری شده محسوب می‌شوند. همچنین امکان خرید هر آهنگ به صورت جداگانه نیز وجود دارد.

وظیفه شما ساده است. با توجه به بودجه  $P$  (مقدار پولی که در اختیار داریم، به تومان)، بیشینه‌ی ارزش موسیقایی کل آهنگ‌هایی را که می‌توانیم بخریم را محاسبه کنید. ارزش موسیقایی کل به عنوان مجموع ارزش‌های موسیقایی تمام آهنگ‌های متمایزی که خریداری شده‌اند (به صورت جداگانه یا به عنوان بخشی از آلبوم‌ها) تعریف می‌شود.

## ورودی

- در اولین خط ورودی شامل سه عدد جدا از هم  $N$ ،  $M$  و  $P$  است (در ابتدا  $N$ ، بعد  $M$  و سپس  $P$  می‌آید).
- $N$  خط بعدی دنبال می‌شود. برای هر  $i$  ( $1 \leq i \leq N$ )، خط  $i$ -ام این خطوط شامل سه عدد جدا از هم  $a_i$ ،  $p_i$  و  $v_i$  است.
- خط آخر شامل  $M$  عدد جدا از هم  $b_1, b_2, \dots, b_M$  است.

## خروجی

یک خط حاوی یک عدد چاپ کنید: بیشینه‌ی ارزش موسیقیایی کل آهنگ‌هایی که می‌توانیم بخریم.

## محدودیت‌ها

- $1 \leq N, M, P \leq 1,000$
- برای هر  $i$  معتبر  $1 \leq b_i, p_i \leq P$
- برای هر  $i$  معتبر  $1 \leq v_i \leq 10^6$
- برای هر  $i$  معتبر  $1 \leq a_i \leq M$

## مثال

### ورودی نمونه ۱

```
5 2 24
1 7 2
1 5 2
1 4 1
2 9 1
2 13 2
10 15
```

### خروجی نمونه ۱

7

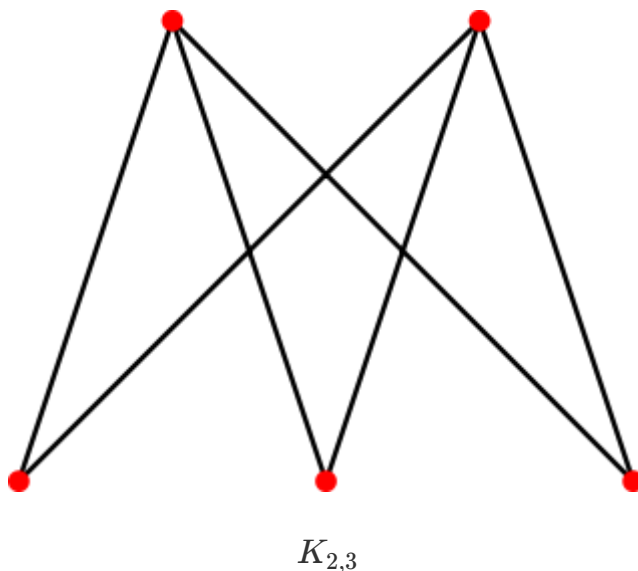
## گرافیابی

- محدودیت زمان: ۲ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

گراف دوبخشی کامل نوع خاصی گراف دوبخشی است. اگر دو بخش رأس‌های این گراف را با  $A$  و  $B$  نمایش دهیم، چنین گراف دوبخشی کاملی با  $K_{|A|,|B|}$  نمایش داده می‌شود و شرایط زیر را برآورده می‌کند:

- به ازای هر  $a \in A$  و  $b \in B$ ، یک یال بین رأس‌های  $a$  و  $b$  وجود دارد.
- به ازای هر  $u, v \in A$ ، هیچ یالی بین این رأس‌ها وجود ندارد.
- به ازای هر  $u, v \in B$ ، هیچ یالی بین این رأس‌ها نیز وجود ندارد.

یک گراف ساده با  $N$  رأس (شماره‌گذاری شده از 1 تا  $N$ ) و  $M$  یال به شما داده می‌شود. (توجه کنید که این گراف لزوماً دوبخشی نیست.) تعیین کنید که آیا این گراف حاوی گراف دوبخشی کامل  $K_{2,3}$  به عنوان زیرگراف هست یا نه. به عبارت دیگر آیا امکان دارد که با حذف برخی (شاید صفر) رأس و برخی (شاید صفر) یال،  $K_{2,3}$  را به دست آوریم یا خیر.



## ورودی

- در اولین خط دو عدد جدا از هم  $N$  و  $M$  قرار دارد (در ابتدا  $N$  می‌آید و سپس  $M$ ).
- در هر کدام از  $M$  خط بعدی دو عدد جدا از هم  $u$  و  $v$  وجود دارد که نشان می‌دهد بین رأس‌های  $u$  و  $v$  یک یال ساده وجود دارد.

## خروجی

یک خط حاوی رشته‌ی YES چاپ کنید اگر گراف حاوی حداقل یک  $K_{2,3}$  باشد و یا NO در صورتی که نباشد (تمامی حروف انگلیسی بزرگ هستند).

## محدودیت‌ها

- $1 \leq N \leq 2000$
- $0 \leq M \leq \frac{N(N-1)}{2}$
- $1 \leq u, v \leq N$
- گراف ورودی ساده است و هیچ طوقه و یال چندگانه‌ای ندارد.

## مثال

### ورودی نمونه ۱

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

خروجی نمونه ۱

YES

## هکاتون

- محدودیت زمان: ۱۰ ثانیه (برای تمامی زبان‌های برنامه‌نویسی)
- محدودیت حافظه: ۵۱۲ مگابایت (برای تمامی زبان‌های برنامه‌نویسی)

شرکت خدمات پیشخوان ایرانیان و شرکت فناوران اطلاعات پیشخوان ایرانیان قصد برگزاری یک مسابقه‌ی برنامه‌نویسی به صورت هکاتون را دارند. در این مسابقه‌ی برنامه‌نویسی، هر تیم روی میز مخصوص به خودش نشسته است. دو شرکتی که اسپانسر برگزاری این هکاتون را دارند نیز، در غرفه‌هایی وسایل و غذای رایگان می‌دهند. سارا و دوستانش نیز در این مسابقه شرکت کرده‌اند.

در حقیقت، سارا برای برنده شدن یا حتی تلاش برای برنده شدن در آنجا نیست؛ او همیشه به دنبال هکاتون‌های جدیدی است که به آن‌ها بپیوندد، به خصوص وقتی که با مزایای رایگان در آن هکاتون‌ها باشن. این بار،  $M$  غرفه‌ی مختلف وجود دارد که شروع به توزیع وسایل و غذای رایگان خواهند کرد. می‌توان گفت هکاتون در یک جدول مستطیلی برگزار می‌شود (یعنی نمای از بالا به پایین مسابقه به این شکل است). غرفه‌ها همگی در سطر اول قرار دارند و غرفه‌ی  $i$ -ام در مختصات  $(B_i, 0)$  قرار دارد. سارا مکان هر یک از  $N$  رقیب دیگر را می‌داند، رقیب  $i$ -ام در مختصات  $(C_i, R_i)$  قرار دارد، در حالی که مکان سارا مختصات  $(X, Y)$  است. ممکن است چند رقیب در یک مختصات قرار گرفته باشند.

هنگامی که توزیع وسایل و غذای رایگان اعلام می‌شود، سارا می‌داند که هر رقیب بلافاصله به نزدیک‌ترین غرفه حرکت خواهد کرد و در صورت پیش آمدن حالات برابر، غرفه‌ی سمت چپ را انتخاب می‌کند. حرکت در شبکه فقط به صورت افقی یا عمودی امکان‌پذیر است و حرکت به اندازه‌ی یک واحد در هر جهت، یک ثانیه طول می‌کشد.

با توجه به این اطلاعات، سارا می‌خواهد استراتژی خود را بهینه کند و احتمال خود را برای انتخاب بهترین وسیله‌ی رایگان موجود بیشینه کند: او به غرفه‌ای می‌رود که کم‌ترین تعداد افراد قبل از او به آنجا برسند. اگر او به غرفه‌ای هم‌زمان با رقیب دیگری برسد، مهربانی او غلبه می‌کند و اجازه می‌دهد آن‌ها اول انتخاب کنند.

به او کمک کنید که تعداد افرادی که قبل یا هم‌زمان با او به غرفه انتخاب شده می‌رسند را محاسبه کند.

## ورودی

- خط اول شامل یک عدد صحیح  $M$  است که تعداد غرفه‌هاست.
- خط دوم شامل  $M$  عدد صحیح  $B_i$  است که هر کدام مختصات ستون  $i$ -امین غرفه هستند.
- خط سوم شامل دو عدد صحیح  $X$  و  $Y$  است که به ترتیب سطر و ستون موقعیت سارا هستند.
- خط چهارم شامل یک عدد صحیح  $N$  است که تعداد سایر رقباست.
- سپس،  $N$  خط بعدی شامل دو عدد صحیح  $R_i$  و  $C_i$  است که به ترتیب سطر و ستون  $i$ -امین رقیب هستند.

## خروجی

شما باید یک خط شامل یک عدد صحیح بنویسید: کمترین تعداد رقیبی که حداقل به همان اندازه زودی که سارا به غرفه برسد، به آن‌جا می‌رسند.

## محدودیت‌ها

- $1 \leq N, M \leq 10^6$
- به ازای تمامی  $i$ ‌های معتبر  $1 \leq B_i \leq 10^9$
- همه‌ی مقادیر  $B_i$  متمایز هستند.
- به ازای تمامی  $i$ ‌های معتبر  $0 \leq R_i, C_i, X, Y \leq 10^9$

## مثال

### ورودی نمونه ۱

```
3
7 3 0
4 0
5
3 1
4 8
```



1 0  
2 3  
3 6

خروجی نمونه ۱

1

توضیح نمونه ۱

