بادکنک‌های مهدی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

چهار ماه دیگر تولد مهدی است. او از الآن تصمیم گرفته که برای جشن تولد خود n بادکنک را به‌صورت خطی در اتاقش بچیند. بادکنک‌ها قرار است با گاز هلیم پر شده و با نخ به زمین متصل شوند. مهدی در نظر دارد تا یک چینش خاص برای این بادکنک‌ها بیابد، به‌طوری که بین هر دو بادکنک با طول نخ (ارتفاع) h، حداقل یک بادکنک با طول نخ کوچک‌تر از h قرار گرفته باشد. البته او یک محدودیت دیگر نیز دارد، و آن این است که حداکثر طول نخ هر بادکنک می‌تواند k باشد.

بر اساس تعداد بادکنک‌ها و حداکثر طول نخ هر بادکنک، یک چینش مناسب برای مهدی ارائه دهید.

ورودی

در یک خط از ورودی، به‌ترتیب دو عدد n و k وارد می‌شود که بیانگر تعداد بادکنک‌ها و حداکثر طول نخ هر بادکنک هستند.

$$1 \leq n, k \leq 10^6$$

خروجی

در یک خط از خروجی، اگر چینش مناسبی برای بادکنک‌ها قابل ارائه بود، آن را چاپ کنید. در غیر این‌صورت، `Impossible` را چاپ کنید.

نکات

- ممکن است بیش از یک چینش برای هر ورودی وجود داشته باشد که همه‌ی آن‌ها قابل قبول هستند.
- طول نخ‌ها باید اعدادی صحیح و بزرگ‌تر از صفر باشند.

مثال

ورودی نمونه ۱

```
5 3
```

خروجی نمونه ۱

```
3 2 3 1 2
```

بین دو بادکنک با ارتفاع ۳، یک بادکنک با ارتفاع ۲ قرار گرفته است. هم‌چنین، بین دو بادکنک با ارتفاع ۲، یک بادکنک با ارتفاع ۱ قرار گرفته است.

ورودی نمونه ۲

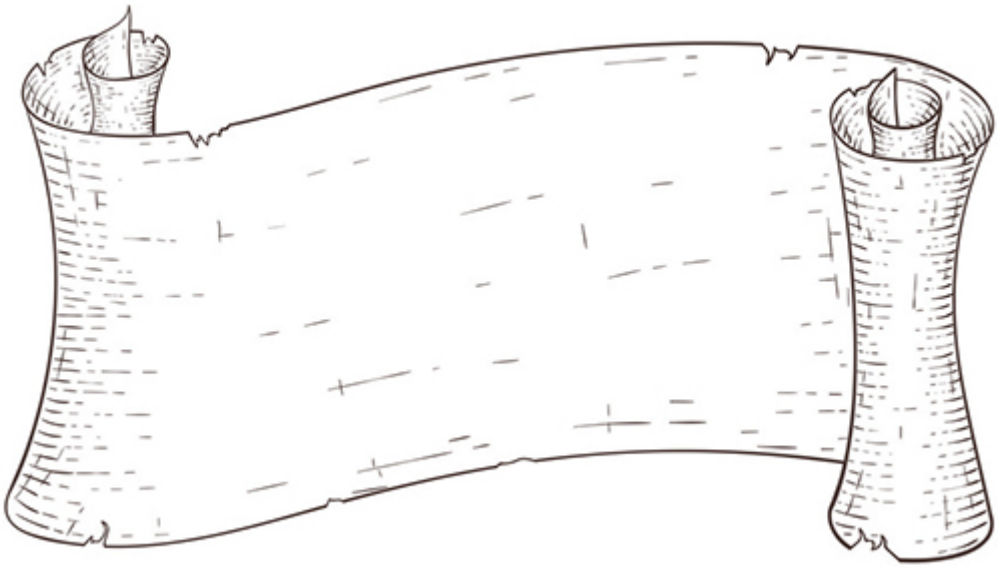
```
6 2
```

خروجی نمونه ۲

```
Impossible
```

بگشای رمز

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



اخیراً مصطفی و پارسا از روشی کاملاً حرفه‌ای برای چت با یکدیگر استفاده می‌کنند. آن‌ها برای انتقال یک پیام، صرفاً ترتیب کلمات را تغییر می‌دهند. مصطفی و پارسا پس از انتقال **خود** کلمات پیام با ترتیب نامعلوم (که شیوه‌ی آن در این مسئله بررسی نمی‌شود)، **ترتیب و جایگشت آن‌ها** را تبدیل به یک رشته‌ی باینری می‌کنند. در حقیقت، رشته‌ی باینری حاوی اطلاعات فرایند مرتب‌سازی ادغامی بر روی جایگشت اولیه و اصلی کلمات است که در نهایت کلمات را به حالت مرتب‌شده در می‌آورد. گیرنده‌ی پیام باید با مهندسی معکوس، جایگشت اولیه‌ی کلمات که در حقیقت اصل پیغام است را بازیابی کند. نحوه‌ی تولید رشته در شبه‌کد زیر توضیح داده شده است:

```
mergeSort(a):
    b = a[0..(a.size / 2)]
    c = a[(a.size / 2)..a.size]
    mergeSort(b)
    mergeSort(c)
    a = merge(b, c)
```

```
merge(a, b):
    merge two sorted arrays a and b. in each step:
        if the element is chosen from a:
            add '0' to the binary string:
        else
            add '1' to the binary string
```

فرستنده در ابتدا تابع `mergeSort` را روی کل آرایه‌ی کلمات (که همان متن اصلی است) صدا زده و در نهایت با کد بالا، رشته‌ی باینری مورد نظر تولید و آماده‌ی ارسال می‌شود. کد رمزنگار، رشته‌ها را با شیوه‌ی خاصی مقایسه و نهایتاً مرتب می‌کند، به این‌صورت که اگر یکی از رشته ها شامل زیررشته‌ی `mosi` باشد، زودتر از رشته‌ی دیگر می‌آید. کوچکی و بزرگی حروف مهم نیست؛ `MosI` یا `mOSI` نیز همین‌طور هستند. اگر هر دو رشته این زیردنباله را داشته باشند یا هر دو نداشته باشند، رشته‌ها به‌صورت *lexicographically* مقایسه می‌شوند؛ به همان سیستمی که در لغت‌نامه‌ها لغت‌ها مرتب شده‌اند. اولویت حروف بزرگ را بالاتر در نظر بگیرید.

مصطفی و پارسا برای گسترش این شیوه‌ی رمزنگاری جهت استفاده در بین سایر دوستان‌شان، از شما می‌خواهند یک برنامه‌ی رمزگشایی برای‌شان بنویسید تا با دریافت یک پیام رمزنگاری‌شده، اصل متن را بازیابی کنید.

ورودی

در خط اول ورودی، عدد صحیح n وارد می‌شود که بیانگر تعداد رشته‌ها است.

$$1 \leq n \leq 10^5$$

در n خط بعدی، رشته‌ها با ترتیبی نامعلوم وارد می‌شوند. این رشته‌ها تنها شامل حروف کوچک و بزرگ انگلیسی هستند. طول هر رشته حداکثر ۵۰ کاراکتر است.

در خط بعدی، رشته‌ی باینری می‌آید.

خروجی

در n خط از خروجی، کلمات متن را با ترتیب اصلی‌شان چاپ کنید.

مثال

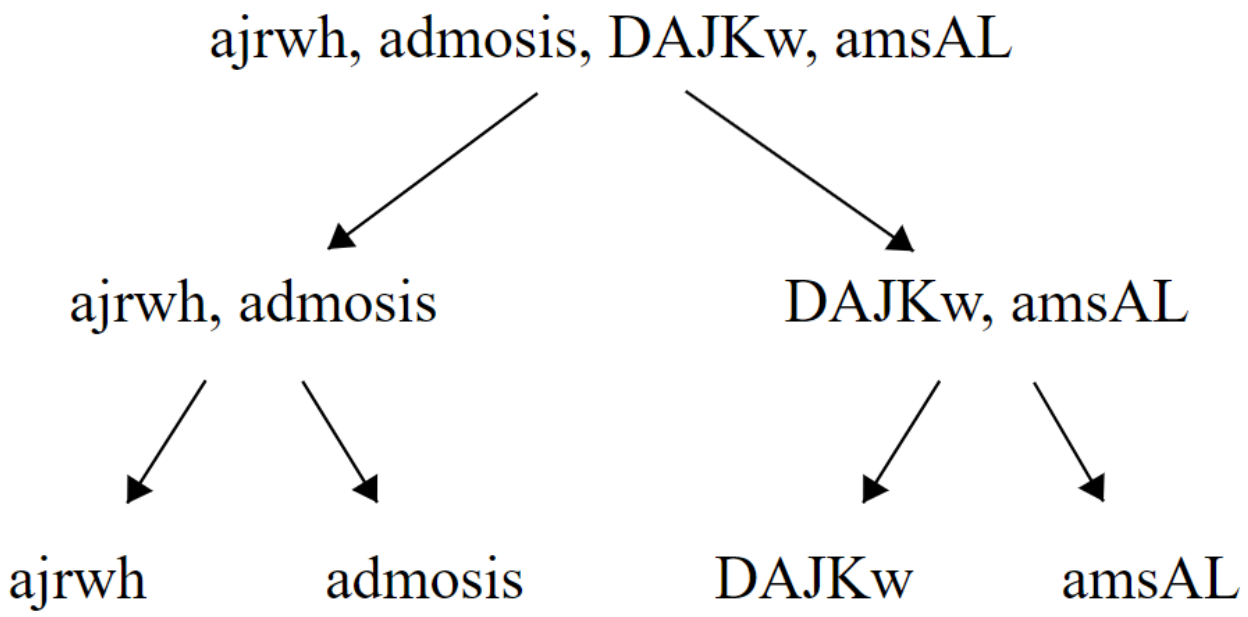
ورودی نمونه ۱

```
4
DAJKw
amsAL
ajrwh
admosis
10010101
```

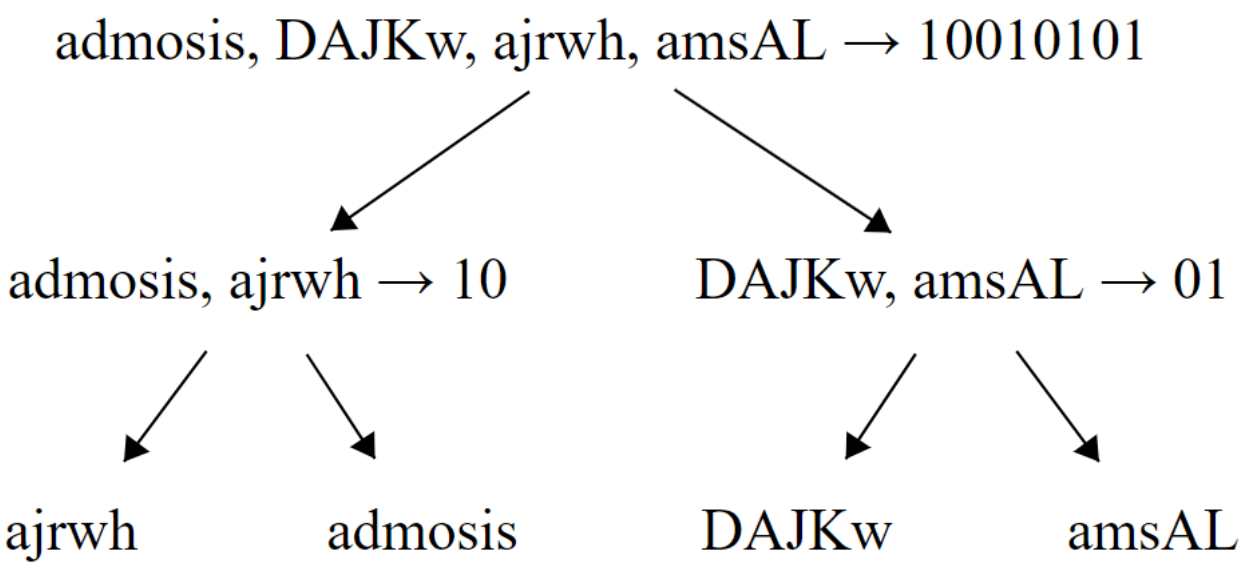
خروجی نمونه ۱

```
ajrwh
admosis
DAJKw
amsAL
```

درخت بازگشت مرتب‌سازی رشته‌ها با شروع از جایگشت اصلی به‌صورت زیر است:



ابتدا دو رشته‌ی `ajrwh` و `admosis` با یکدیگر مقایسه می‌شوند. از آن‌جایی که `admosis` شامل زیررشته‌ی `mosi` است، باید قبل از `ajrwh` بیاید. بنابراین، دو بیت ابتدایی رشته‌ی باینری برابر با `10` است. در ادامه، دو رشته‌ی `DAJKw` و `amsAL` با یکدیگر مقایسه می‌شوند. از آن‌جایی که هیچ‌یک از این دو رشته شامل زیررشته‌ی `mosi` نیستند، باید به‌صورت *lexicographically* مقایسه شوند. `D` اولویت بیش‌تری نسبت به `a` دارد، بنابراین `DAJKw` باید قبل از `amsAL` بیاید. پس، بیت‌های سوم و چهارم رشته‌ی باینری برابر با `01` هستند. در نهایت، باید عملیات ادغام را روی دو بخش `admosis,ajrwh` و `DAJKw,amsAL` انجام داد. ابتدا `admosis` می‌آید، پس بیت بعدی `0` است. سپس `DAJKw` می‌آید، پس بیت بعدی `1` است. در ادامه، `ajrwh` می‌آید، پس بیت بعدی `0` است. در نهایت `amsAL` می‌آید، پس آخرین بیت `1` است. رشته‌ی باینری به‌دست آمده، همان رشته‌ی باینری ورودی است.



ورودی نمونه ۲

```
7
ACSmosicLm
mOSICVKnHLYJs
NHCpGb
uRcGD
ZxMOsIC
cUUbXcf
RdEt
01110010101010010111
```

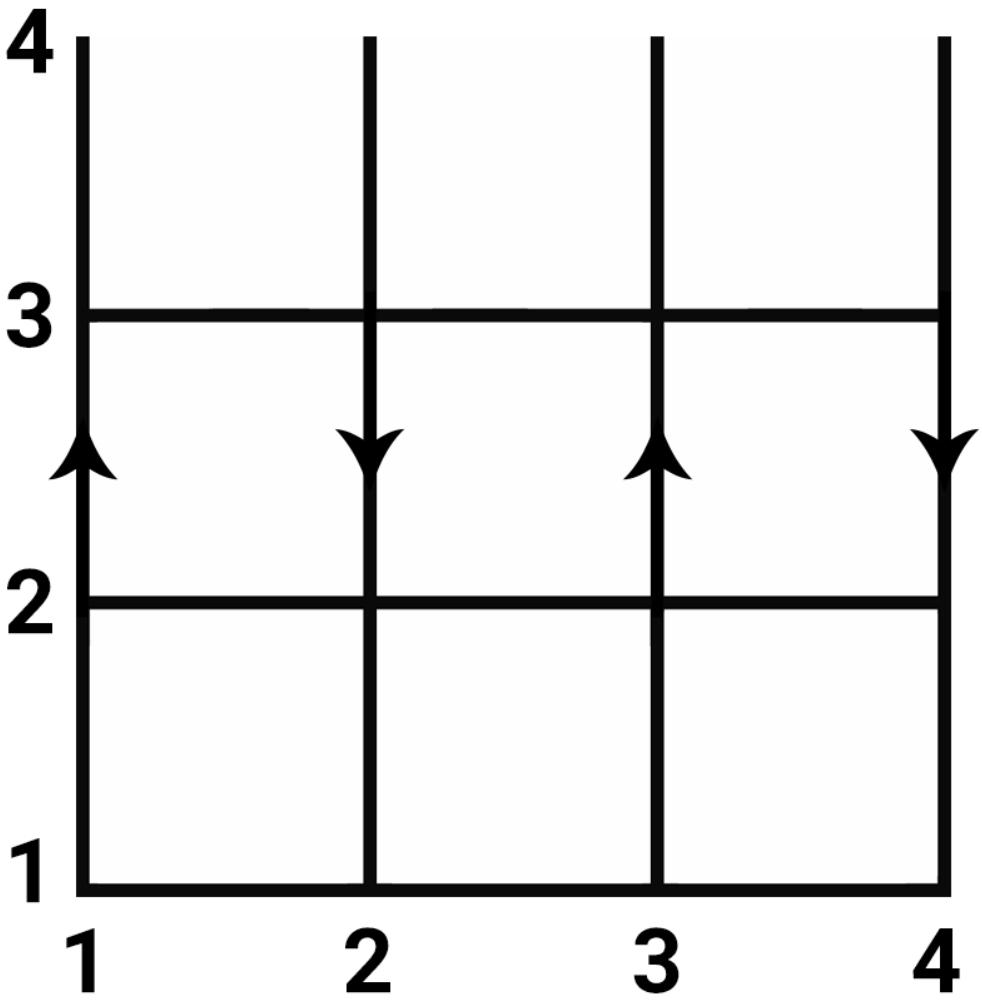
خروجی نمونه ۲

```
NHCpGb
ACSmosicLm
ZxMOsIC
mOSICVKnHLYJs
cUUbXcf
RdEt
uRcGD
```

دوربرگردان

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در شهر *کدنشین‌ها* چهار خیابان یک‌طرفه وجود دارد که اگر آن‌ها را روی جدول مختصات در نظر بگیریم، به شکل زیر هستند:



خیابان‌های ۱ و ۳ رو به بالا و خیابان‌های ۲ و ۴ رو به پایین هستند. این خیابان‌ها از بالا نامتناهی هستند. همچنین، بین خیابان‌های این شهر، تعدادی دوربرگردان **دوطرفه**(!) وجود دارد.

نیما هم‌اکنون در نقطه‌ای با مختصات $(1, s)$ قرار دارد و می‌خواهد به نقطه‌ی $(4, f)$ برود. هر واحد حرکت ۱ ثانیه زمان می‌برد. عبور از هر دوربرگردان نیز ۱ ثانیه زمان می‌برد. به نیما بگویید که اگر کوتاه‌ترین مسیر را طی کند، چند ثانیه طول می‌کشد تا به مقصدش برسد.

ورودی

در خط اوّل ورودی استاندارد، عدد طبیعی s نوشته می‌شود که مؤلفه‌ی y مکان اولیه‌ی نیما است.

$$1 \leq s \leq 10^9$$

در خط دوم، عدد طبیعی f نوشته می‌شود که مؤلفه‌ی y مقصد نیما است.

$$1 \leq f \leq 10^9$$

در خط سوّم، عدد طبیعی n نوشته می‌شود که بیانگر تعداد دوربرگردان‌ها است.

$$1 \leq n \leq 10^5$$

در n خط بعدی، در هر خط سه عدد طبیعی x_i و x_j و y_i نوشته می‌شود که به‌ترتیب بیانگر مختصات x دو طرف دوربرگردان و مؤلفه‌ی y دوربرگردان است.

$$1 \leq x_i, x_j \leq 4$$

$$|x_i - x_j| = 1$$

$$1 \leq y_i \leq 10^9$$

خروجی

در یک خط از خروجی استاندارد، کمینه‌ی زمان موردنیاز برای رسیدن نیما به مقصدش را بر حسب ثانیه چاپ کنید. اگر مسیری برای رسیدن به مقصد وجود نداشت، `-1` را چاپ کنید.

مثال

ورودی نمونه ۱

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

خروجی نمونه ۱

```
11
```

ورودی نمونه ۲

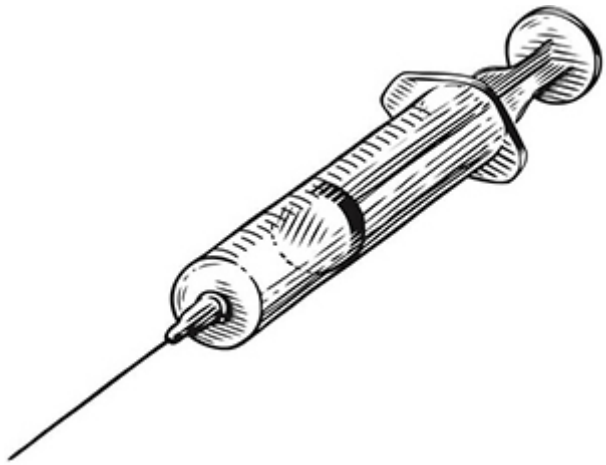
```
2
11
8
1 2 7
1 2 1
2 1 5
2 3 8
3 2 5
2 3 6
2 3 9
3 4 10
```

خروجی نمونه ۲

```
-1
```

واکسیناسیون فوری

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت



در سرزمین *سیلیکون‌ولی‌زادگان* واکسیناسیون کرونا آغاز شده است. از آن‌جایی که سلامتی مردمان *سیلیکون‌ولی‌زادگان* برای دولت این سرزمین بسیار مهم است، در مجلس این کشور لایحه‌ای مبنی بر کمینه بودن زمان اتمام واکسیناسیون تصویب شده است. بر این اساس، تعدادی ایستگاه واکسیناسیون در شهرهای مختلف احداث شده است. از آن‌جایی که اعضای مجلس این دولت اکثراً *computer scientist* هستند، شهرهای این کشور را به شکل یک گراف در آورده‌اند. در این گراف، همه‌ی شهرها مشخص شده‌اند. هر شهر ممکن است در وضعیت قرمز باشد. هم‌چنین، در هر شهر می‌تواند ایستگاه‌های واکسیناسیون احداث شده باشد. حال، دولت *سیلیکون‌ولی‌زادگان* می‌خواهد نزدیک‌ترین ایستگاه واکسیناسیون به هر شهری که در وضعیت قرمز است و فاصله‌ی شهر تا ایستگاه واکسیناسیون را به‌دست آورد. آن‌ها را در تحقق این امر یاری کنید.

ورودی

در خط اول، دو عدد صحیح n و m وارد می‌شود که به‌ترتیب بیانگر تعداد شهرها و تعداد جاده‌ها هستند.

$$1 \leq n, m \leq 10^5$$

در n خط بعدی، در هر دو عدد صحیح p_i و q_i وارد می‌شود. p_i می‌تواند صفر یا ۱ باشد که اگر ۱ باشد، به این معناست که وضعیت شهر iاُم قرمز است. q_i نیز می‌تواند صفر یا ۱ باشد که اگر ۱ باشد، به این معناست که در شهر iاُم ایستگاه واکسیناسیون احداث شده است.

در m خط بعدی، در هر خط سه عدد صحیح u و v و w وارد می‌شود که به‌ترتیب بیانگر شناسه‌ی عددی شهرهای دو سر جاده و طول جاده است. همه‌ی جاده‌ها دوطرفه هستند.

$$1 \leq u, v \leq n$$

$$1 \leq w \leq 10^9$$

خروجی

در x خط از خروجی (x تعداد شهرهایی است که وضعیت‌شان قرمز است)، در هر خط دو عدد چاپ کنید به‌طوری که اولین عدد به‌معنای شناسه‌ی عددی شهری که وضعیتش قرمز است و دومین عدد به‌معنای فاصله‌ی این دو شهر از یکدیگر باشد. اگر شهر به ایستگاه واکسیناسیون دسترسی نداشت، به جای مقدار فاصله، `-1` را چاپ کنید.

نکته: فاصله‌ی شهرهای خروجی باید به‌ترتیب صعودی بر حسب شناسه باشد.

مثال

ورودی نمونه

```
5 4
1 0
1 1
0 1
```

```
0 0
0 0
1 2 4
3 4 1
3 2 5
4 5 1
```

خروجی نمونه

```
1 4
2 0
```

مأموریت احتمالاً ممکن

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

محمدرضا اخیراً مدیر فنی شرکت *کدنویس‌گستران غرب به‌جز نیما* شده است. این شرکت اخیراً به یک تکنولوژی پیشرفته دست یافته و قصد دارد این تکنولوژی را به تعدادی استارت‌آپ بفروشد. برای فروش این تکنولوژی، محمدرضا باید از شهری که در آن قرار دارد به هرکدام از شهرهای مدنظرش سفر کند و در نهایت نیز به شهر خودش برگردد. او برای سفرهای بین شهری می‌تواند از ایرلاین های موجود استفاده کند. ایرلاین iاُم از شهر u_i به شهر v_i در مدت w_i دقیقه پرواز می‌کند (پروازها در هر دو مسیر رفت و برگشت موجود هستند؛ بنابراین، از پرواز iاُمین ایرلاین می‌توان برای سفر از شهر u_i به شهر v_i در مدت w_i دقیقه استفاده کرد).

از آن‌جا که محمدرضا باید بر روی کارهای شرکت نظارت کند، نمی‌تواند بیش‌تر از t دقیقه برای فروش این تکنولوژی وقت صرف کند. هم‌چنین، محمدرضا یک جت شخصی دارد که می‌تواند حداکثر یک بار از آن استفاده کند. این جت مسیر بین هر دو شهر دلخواهی را در مدت j دقیقه طی می‌کند، اما از آن‌جا که استفاده از جت شخصی برای محمدرضا هزینه‌بر است، اگر محمدرضا بتواند در زمانی کم‌تر یا مساوی t مأموریتش را انجام دهد و به شرکت باز گردد، از جتش استفاده نمی‌کند.

به محمدرضا بگویید آیا می‌تواند این مأموریت را در حداکثر t دقیقه به پایان برساند یا خیر.

محمدرضا سفر خود را از شهر ۰ آغاز می‌کند.

ورودی

در خط اول ورودی، به‌ترتیب پنج عدد صحیح n و m و k و t و j وارد می‌شود. n بیانگر تعداد شهرها، m بیانگر تعداد شهرهای مدنظر محمدرضا، k بیانگر تعداد ایرلاین‌ها، t بیانگر حداکثر زمانی است که محمدرضا می‌تواند برای فروش تکنولوژی صرف کند و j بیانگر زمان سفر با جت است.

$$1 \leq n \leq 2 \times 10^4$$

$$1 \leq m \leq 15$$

$$1 \leq k, t \leq 10^5$$

$$1 \leq j \leq 500$$

در m خط بعدی، در هر خط دو عدد صحیح c_i و s_i وارد می‌شود که به‌ترتیب بیانگر شماره‌ی شهر مدنظر محمدرضا و مدت زمانی است که قرار است در آن‌جا بماند است. شماره‌ی شهرها از **صفر** آغاز می‌شود.

$$0 \leq c_i \leq n$$

$$1 \leq s_i \leq 500$$

در k خط بعدی، در هر خط سه عدد صحیح u_i و v_i و w_i وارد می‌شود که به‌ترتیب بیانگر شماره‌ی شهرهای دو سر ایرلاین و مدت زمان سفر هستند.

$$0 \leq u_i, v_i \leq n$$

$$1 \leq w_i \leq 500$$

خروجی

در یک خط از خروجی استاندارد، در صورتی که امکان انجام مأموریت بدون استفاده از جت وجود دارد `YES` ، اگر امکان انجام مأموریت با استفاده از جت وجود دارد `YES+JET` و اگر امکان انجام مأموریت در زمان گفته‌شده وجود ندارد، `NO` را چاپ کنید.

مثال

ورودی نمونه ۱

```
6 3 10 18 5
1 2
4 2
5 2
0 1 2
1 2 3
2 4 3
1 3 10
2 3 6
0 3 2
3 4 2
4 5 1
3 5 2
0 5 5
```

خروجی نمونه ۱

```
YES+JET
```

ورودی نمونه ۲

```
2 2 2 10 500
0 1
1 1
0 1 1
0 1 10
```

خروجی نمونه ۲

```
YES
```

ورودی نمونه ۳

```
6 2 7 99 1
2 1
4 1
0 1 1
0 2 1
1 2 1
1 3 100
3 4 1
3 5 1
4 5 1
```

خروجی نمونه ۳

```
NO
```

تکرار سؤال اول تمرین قبل

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

علیش که متوجه شده نه اسم‌های سؤالات و نه داستان‌های ابتدای سوال جذابیتی برای کسی ندارد، تصمیم گرفته تا این سؤال را مستقیما بیان کند! سوال به این‌گونه است که n صندوق داریم که روی یک خط کنار هم قرار گرفته‌اند و از چپ به راست از ۱ تا n شماره‌گذاری شده‌اند.

هر صندوق فقط با کلید مخصوص خود باز می‌شود و داخل هر صندوق کلید صندوق چپ و راستش (در صورت داشتن چپ و راست) قرار دارد.

داخل بعضی از صندوق‌ها توپ است و ما k کلید در دست داریم و باید با کم‌ترین تعداد صندوقی که باز می‌کنیم همه‌ی توپ ها را برداریم.

ورودی

در خط اول ورودی، سه عدد n و k و m که به‌ترتیب بیانگر تعداد صندوق‌ها، تعداد کلیدها و تعداد توپ‌ها هستند وارد می‌شود.

در خط دوم k عدد وارد می‌شود که iاُمین آن‌ها بیانگر این است کلید i چه صندوقی را باز می‌کند (شماره‌ی صندوق وارد می‌شود).

در خط سوم m عدد داده می‌شود که iاُمین آن‌ها بیانگر این است که توپ iاُم در کدام صندوق قرار گرفته است.

$$1 \leq n \leq 500$$

$$1 \leq k \leq 500$$

$$0 \leq m \leq 500$$

خروجی

در تنها خط خروجی، مینیمم جواب مسئله را چاپ کنید.

ورودی نمونه ۱

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

خروجی نمونه ۱

```
5
```

ورودی نمونه ۲

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

خروجی نمونه ۲

```
4
```