

تبدیل گراف به درخت

- محدودیت زمان: ۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک گراف ساده (گرافی بدون جهت و بدون وزن و بدون یال چندگانه و بدون طوقه) به شما داده می‌شود. شما باید کمترین تعداد ممکن یال را از گراف حذف کنید و کمترین تعداد ممکن یال 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.

$$1 \leq n \leq 2000$$

$$0 \leq m \leq n * (n - 1) / 2$$

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، a ، تعداد یال‌های لازم برای حذف شدن از گراف را 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 $a + 2$ امین خط، b ، تعداد یال‌های لازم برای اضافه شدن به گراف را چاپ کنید. در هر یک از b خط بعدی، شماره‌ی رئوس دو سر هر یالی که لازم است از گراف حذف 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
1 3
0
```

ورودی نمونه ۳

4 0

خروجی نمونه ۳

0
3
1 2
2 3
3 4

نکات

- پیشنهاد می‌شود از الگوریتم DFS یا BFS استفاده کنید. برای یاد گرفتن تئوری این مبحث، جست‌وجوی زیاد در اینترنت و استفاده از گوگل اکیداً توصیه می‌شود (هرچند کلیت موضوع در کلاس حل تمرین گفته شده‌است)، اما از کپی کردن کدها (به دلیل سخت‌گیری در عملیات گرفتن تقلب‌ها) خودداری کنید.
- 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 حذف شده و یال‌های اضافه شده باید کمینه باشد.
- کد این سوال را با یکی از زبان‌های C و C++ و Java و Python می‌توانید بزنید اما دقت کنید که در هر حالت، کد شما زیر ۵ ثانیه باید اجرا شود. به دلیل زیاد بودن محدودیت زمانی، اعتراضات درمورد آن پذیرفته نمی‌باشد.

اسب!

- محدودیت زمان: ۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک مهره‌ی اسب روی یک صفحه‌ی شطرنج n در m قرار گرفته‌است. تعدادی از خانه‌ی صفحه‌ی شطرنج پر هستند و بقیه خالی هستند. هر بار می‌توانیم اسب را از جای خود جابه‌جا کنیم و طبق [حرکات استاندارد](#) آن به یک خانه‌ی خالی دیگر منتقل کنیم. هدف این است که مهره‌ی اسب را از مکان اولیه‌اش به یکی از [مکان‌های نهایی](#) مدنظرمان منتقل کنیم، طوری که تعداد کل جابه‌جایی‌ها کمینه شود.

در نظر داشته‌باشید که سطرهای صفحه‌ی شطرنج از بالا به پایین با اعداد طبیعی 1 تا n شماره‌گذاری شده‌اند و ستون‌های آن از چپ به راست با اعداد طبیعی 1 تا m شماره‌گذاری شده‌اند.

ورودی

در اولین خط ورودی دو عدد n و m با یک فاصله بینشان آمده‌است که به ترتیب تعداد سطرها و ستون‌های صفحه‌ی شطرنج را نشان می‌دهد.

$$2 \leq n, m \leq 1000$$

در ادامه n خط در ورودی آمده‌است که هر کدام دقیقاً m کاراکتر دارد. این کاراکترها صفحه‌ی شطرنج را توصیف می‌کنند و از چهار حالت زیر خارج نیستند:

- کاراکتر K (حرف بزرگ) نشان‌دهنده‌ی مکان اولیه‌ی مهره‌ی اسب است.
- کاراکتر D (حرف بزرگ) نشان‌دهنده‌ی یک مکان نهایی مناسب برای مهره‌ی اسب است.
- کاراکتر $-$ نشان‌دهنده‌ی یک خانه‌ی خالی روی صفحه‌ی شطرنج است (که اسب می‌تواند روی آن قرار گیرد).
- کاراکتر $\#$ نشان‌دهنده‌ی یک خانه‌ی پر روی صفحه‌ی شطرنج است (که اسب هیچ‌گاه نمی‌تواند روی آن قرار گیرد).

ورودی شامل دقیقاً یک کاراکتر K و حداقل یک کاراکتر D است.

خروجی

در اولین خط، a کمینه‌ی تعداد حرکت لازم برای مهره‌ی اسب را چاپ کنید. در $a + 1$ خط بعدی مسیر حرکت مهره‌ی اسب را برای رسیدن به نزدیک‌ترین مقصد نهایی خود را به ترتیب توصیف کنید. در هر یک از $a + 1$ خط بعدی، دو عدد طبیعی که نشان‌دهنده‌ی سطر و ستون خانه‌ای که قرار است اسب در آن قرار گیرد را با یک فاصله بینشان چاپ کنید. دقت کنید که مسیر باید از خانه‌ی مبدأ شروع شود و به یکی از خانه‌های مقصد ختم شود. تمام خانه‌های میانی مسیر نیز باید خالی باشند. همچنین دقت کنید که خود خانه‌ی اولیه و نهایی مسیر را هم چاپ کنید.

اگر امکان رسیدن به خانه‌های مقصد وجود نداشت، در یک خط فقط -1 چاپ کنید.

مثال

ورودی نمونه ۱

```
4 3
KDD
###
###
###
```

خروجی نمونه ۱

```
-1
```

ورودی نمونه ۲

```
4 3
KDD
###
```

```
#-#
###
```

خروجی نمونه ۲

```
2
1 1
3 2
1 3
```

ورودی نمونه ۳

```
4 3
---
---
---
DDK
```

خروجی نمونه ۳

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

نکات

- پیشنهاد می‌شود از الگوریتم BFS استفاده کنید. برای یاد گرفتن تئوری این مبحث، جست‌وجوی زیاد در اینترنت و استفاده از گوگل اکیداً توصیه می‌شود (هرچند کلیت موضوع در کلاس حل تمرین گفته شده‌است)، اما از کپی کردن کدها (به دلیل سخت‌گیری در عملیات گرفتن تقلب‌ها) خودداری کنید.
- هر جواب دلخواهی که اسب را از خانه‌ی مبدأ به یکی از خانه‌های مقصد برساند معتبر است. فقط دقت کنید که مقدار عدد طبیعی n (طول مسیر) باید کمینه باشد.
- کد این سوال را با یکی از زبان‌های C و C++ و Java و Python می‌توانید بزنید اما دقت کنید که در

هر حالت، کد شما زیر ۵ ثانیه باید اجرا شود. به دلیل زیاد بودن محدودیت زمانی، اعتراضات درمورد آن پذیرفته نمی‌باشد.