

نارنگی

• محدودیت زمان: ۱ ثانیه (پایتون: ۱۰ ثانیه)

• محدودیت حافظه: ۵۱۲ مگابایت

محسن چند سالی است که مدرک کارشناسی مهندسی کامپیوترش را گرفته است، ولی از آنجایی که به سربازی نرفته، نمیتواند در شرکتی استخدام شود. به همین دلیل برای درآوردن خرج خانواده‌اش در باغ عمویش کار می‌کند. این باغ به شکل یک جدول n در n است که در سطر i ام و ستون j ام آن یک درخت نارنگی وجود دارد که $a_{i,j}$ تا نارنگی دارد.

در یکی از این روزها، محسن که از غم روزگار بسیار خسته شده بود، از بالای باغ وارد شد و تلو تلو خوران می‌خواست خودش را به لب رودی که در پایین باغ جریان داشت برساند. در هر مرحله اگر محسن در خانه i, j (سطر i ام و ستون j ام) است، یا به خانه $i + 1, j + 1$ (در صورت وجود) یا به خانه $i + 1, j - 1$ (در صورت وجود) می‌رود. علاوه بر این، او در طول مسیر حداکثر k بار می‌تواند حواسش را جمع کند و یک خانه‌ی مستقیم به پایین (یعنی به خانه $i + 1, j$) برود. او به هر خانه‌ای که می‌رسد نارنگی‌های روی درخت آن را می‌شمارد. برنامه‌ای بنویسید که حداکثر تعداد نارنگی‌هایی را که محسن می‌تواند در طول مسیر دیده باشد پیدا کند.

ورودی

در خط اول ورودی، n و k داده شده است. در n خط بعدی، در هر خط n عدد داده شده است که عدد j ام در سطر i ام نشان‌دهنده‌ی تعداد نارنگی‌های آن خانه یا همان $a_{i,j}$ است.

$$2 \leq n \leq 100$$

$$0 \leq k \leq 100$$

$$0 \leq a_{i,j} \leq 100$$

نکته: ۳۰ نمره به تست‌های حالت $k = 0$ و ۳۰ نمره به تست‌های با شرط $n, k \leq 50$ اختصاص یافته است.

خروجی

در تنها خط خروجی حداکثر تعداد نارنگی‌هایی که محسن می‌تواند در طول مسیر دیده باشد را چاپ کنید.
مسیر از یک خانه‌ی دلخواه از سطر اول آغاز شده و به یک خانه‌ی دلخواه از سطر آخر ختم می‌شود.

مثال

ورودی نمونه ۱

```
3 1
0 10 0
0 10 0
10 11 0
```

خروجی نمونه ۱

30

توضیح نمونه ۱

محسن می‌تواند از ستون دوم سطر اول شروع کند و سپس از ۱ باری که می‌تواند حواسش را جمع کند استفاده کرده و به خانه‌ی پایینش برود و سپس به خانه‌ی پایین چپش برود. و بدین ترتیب در مجموع ۳۰ عدد نارنگی را ببیند.

مصائب MST

- محدودیت زمان: ۲ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

پوپک که به تازگی در درس طراحی الگوریتم‌ها با درخت پوشای کمینه (MST) آشنا شده است، می‌خواهد بفهمد که چقدر درس را فهمیده است. به همین دلیل از آقا تورج -که از اساتید کار با MST است- می‌خواهد که سوالی در مورد MST به او بدهد.

آقا تورج یک گراف ساده و همبند n رأسی و m یالی به پوپک می‌دهد. پوپک خیلی سریع MST این گراف را حساب کرده و به آقا تورج می‌دهد. آقا تورج لبخندی زده و می‌گوید این که خیلی ساده است! بنابراین او از پوپک می‌خواهد که مسئله‌ای سخت‌تر حل کند. آقا تورج از پوپک می‌خواهد که بررسی کند برای هر مجموعه از یال‌ها که او بگوید، آیا MST ای وجود دارد که شامل همه آن یال‌ها باشد یا خیر. پوپک با دیدن این سوال به نظرش رسیده که متأسفانه خیلی خوب MST را نیاموخته، به همین جهت دل شکسته شده است و از شما کمک می‌خواهد تا پاسخ سوالات آقا تورج را بدهید.

ورودی

خط اول ورودی شامل دو عدد n و m است که به ترتیب تعداد رأس‌ها و تعداد یال‌های گراف را مشخص می‌کند. سپس در m سطر بعدی در هر سطر سه عدد u_i, v_i, w_i می‌آید که بیانگر یک یال میان u_i و v_i با وزن w_i است. سپس در خط بعدی یک عدد q می‌آید که تعداد پرسش‌های آقا تورج است. در q سطر بعدی، در ابتدا یک c_i می‌آید که تعداد یال‌های درون این پرسش را معلوم می‌کند. سپس c_i عدد در ادامه همان سطر آمده است که شماره یال‌هایی است که آقا تورج می‌خواهد بداند آیا MST ای وجود دارد که شامل همه آن یال‌ها باشد یا خیر (شماره یال‌ها عددی در بازه 1 تا m است که در هر پرسش متمایز است).

- $1 \leq n, m \leq 3 \times 10^5$
- $n - 1 \leq m$
- $1 \leq c_i \leq n - 1, \sum_{i=1}^q c_i \leq 3 \times 10^5$
- $1 \leq u_i, v_i \leq n$

$$\bullet 1 \leq w_i \leq 2 \times 10^5$$

خروجی

خروجی باید شامل q سطر باشد که در سطر i -ام آن، چنانچه MST ای موجود باشد که شامل یال‌های مورد نظر آقا تورج در پرسش i -ام باشد، YES چاپ شود وگرنه NO چاپ بشود.

راهنمایی

الگوریتم کروسکال را در نظر بگیرید، فرض کنید گام به گام این الگوریتم را اجرا کرده‌ایم. اکنون یک درخواست در نظر بگیرید و فرض کنید مشتکل از یال‌های e_1, e_2, \dots, e_k باشد. فرض کنید در این درخواست، یال‌های $e_{i_1}, e_{i_2}, \dots, e_{i_s}$ همگی وزن w داشته باشند. اکنون تا مرحله‌ای از الگوریتم کروسکال را در نظر بگیرید که همه یال‌ها با وزن کمتر از w را اضافه کرده باشیم. اگر یال‌های این درخواست بتوانند در یک MST ظاهر شوند، لازم است بتوانیم همه یال‌های $e_{i_1}, e_{i_2}, \dots, e_{i_s}$ را در وضعیت فعلی الگوریتم کروسکال اضافه کنیم و همه این یال‌ها دو مولفه را ادغام (Merge) کنند.

پس شرط لازم و کافی این است که برای هر درخواست یال‌ها را براساس وزن جدا کنیم و برای هر دسته از یال‌های هم‌وزن آن درخواست، بررسی کنیم که در الگوریتم کروسکال تا مرحله کم‌وزن‌تر از وزن آن یال‌ها، آیا این یال‌ها مولفه‌های متفاوت ادغام می‌کنند یا نه.

مثال

ورودی نمونه

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

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

خروجی نمونه

YES
 YES
 YES
 NO
 NO
 YES
 YES
 NO
 NO
 NO

گراف بدون جهت

• محدودیت زمان: ۱ ثانیه (پایتون: ۳ ثانیه)

• محدودیت حافظه: ۵۱۲ مگابایت

پیمان یک گراف جهت‌دار به پوران هدیه داده است. اما از آن جایی که پوران از گراف جهت‌دار خوشش نمی‌آید، می‌خواهد تمامی یال‌های آن را پاک کند تا به گرافی بدون جهت تبدیل شود.

پوران در هر عملیات می‌تواند یک رأس را انتخاب کند و تمامی یال‌های ورودی یا یال‌های خروجی آن را حذف کند. برای رأس شماره‌ی i هزینه‌ی حذف یال‌های ورودی in_i و هزینه‌ی حذف یال‌های خروجی out_i است. پوران از شما می‌خواهد راهی با کم‌ترین هزینه برای پاک کردن تمام یال‌های گراف را بیابید.

ورودی

در خط اول ورودی، n و m داده شده است که نمایانگر تعداد رأس‌ها و یال‌ها است.

در خط دوم ورودی، n عدد داده شده که عدد i ام نمایانگر in_i است.

در خط سوم ورودی، n عدد داده شده که عدد i ام نمایانگر out_i است.

در خط i ام از m خط بعدی، در هر خط دو عدد u_i و v_i آمده که نشان می‌دهد یک یال جهت‌دار از u_i به v_i در گراف وجود دارد. گراف ورودی می‌تواند طوقه یا یال چندگانه داشته باشد.

$$1 \leq n \leq 100$$

$$1 \leq m \leq 5000$$

$$1 \leq in_i, out_i \leq 10^6$$

خروجی

در خط نخست خروجی، مجموع هزینه‌ی مورد نیاز برای حذف یال‌ها را چاپ کنید.

در خط دوم عدد k ، تعداد عملیات‌ها، را چاپ کنید و در k خط بعدی، در هر خط ابتدا شماره‌ی رأس و سپس جهت یال‌های حذف‌شده را چاپ کنید. اگر یال‌های ورودی حذف شده‌اند کلمه‌ی in و اگر یال‌های خروجی حذف شده‌اند کلمه‌ی out را چاپ کنید.

اگر چند جواب بهینه وجود داشت، کافی است یکی را به دلخواه چاپ کنید.

راهنمایی

مسئله را می‌توان به این شکل دید که به ازای هر رأس دو دکمه وجود دارد: یکی برای حذف یال‌های ورودی و حذف یال‌های خروجی. چون چند باز زدن یک دکمه فایده‌ای ندارد پس هر دکمه‌ای را یا می‌زنیم یا نمی‌زنیم. حال اگر وضعیت زده شدن دکمه‌ها را داشته باشیم به ازای هر یال اگر دکمه‌ی ورودی رأس انتهایی یال یا دکمه‌ی خروجی ابتدای یال زده شده باشد، می‌توان فهمید که همه یال‌ها حذف شده است.

از طرفی چون می‌خواهیم هزینه را مینیمم کنیم باید گرافی بسازید که min-cut در آن برابر با هزینه حذف یال‌ها باشد. (گرافی دو بخشی که در یک طرف دکمه‌های ورودی قرار دارد و در طرف دیگر دکمه‌های خروجی و یال‌های گراف بین این دو بخش قرار دارد. source و sink نیز هر یک به یکی از این دو بخش وصل شده اند).

حال با پیدا کردن یال‌های min-cut می‌توانید مشخص کنید که دقیقا کدام دکمه‌ها زده می‌شوند.

مثال

ورودی نمونه ۱

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

خروجی نمونه ۱

5

3

1 in

2 out

2 in