

## انبار غله (Rice Hub)

در حومه‌ی شهر، جاده‌ی مستقیمی به نام «جاده‌ی برنج» وجود دارد. در طول این جاده  $R$  مزرعه‌ی برنج وجود دارند. هر کدام از این مزارع یک مکان صحیح و یک‌بُعدی بین ۱ تا  $L$  (شامل خود این مقادیر) دارد. این مزارع برنج در یک لیست غیرنزولی برحسب مکان‌شان داده شده‌اند. به بیان دقیق‌تر، برای هر اندیس  $0 \leq i < R$ ، مزرعه‌ی برنج شماره  $i$  در مکان  $X[i]$  قرار دارد. می‌توانید فرض کنید که  $1 \leq X[0] \leq \dots \leq X[R-1] \leq L$  است. توجه داشته باشید که چند مزرعه‌ی متفاوت برنج ممکن است در یک مکان یکسان واقع شده باشند.

ما قصد داریم یک انبارغله احداث کنیم تا حداکثر محصول ممکن از مزارع برنج را در انبارغله جمع کنیم. مشابه مزارع برنج، این انبارغله هم باید در مکانی صحیحی بین ۱ تا  $L$  (شامل خود این اعداد) تأسیس شود. این انبار در هر مکانی می‌تواند تأسیس شود، حتی مکانی که در حال حاضر شامل یک مزرعه است هم می‌تواند محل تأسیس انبار باشد.

هر مزرعه‌ی برنج در فصل زراعی، به‌اندازه‌ی دقیقاً یک کامیون پُر، محصول تولید می‌کند. برای انتقال این برنج‌ها به انبارغله، مسئولین شهر می‌بایست یک راننده‌ی کامیون استخدام کنند. این راننده برای انتقال یک کامیون پر از برنج از یک مزرعه به سمت انبار غله، به‌ازای هر یک واحد طول، یک بت تایلندی حق‌الزحمه می‌گیرد. به بیان دیگر، هزینه‌ی انتقال برنج از یک مزرعه داده شده تا انبارغله از نظر عددی برابرست با فاصله‌ی بین آن دو نقطه.

متأسفانه بودجه‌ی ما برای این فصل محدود است؛ ما حداکثر  $B$  بت می‌توانیم خرج این نقل و انتقالات کنیم. وظیفه‌ی شما این است که به ما کمک کنید تا محل تأسیس انبار غله را طوری برگزینیم که با بودجه‌ی محدودمان بتوانیم حداکثر میزان برنج در انبار قابل جمع کنیم.

### وظیفه شما

شما باید روال  $\text{besthub}(R, L, X, B)$  را بنویسید که پارامترهای زیر را بگیرد:

- $R$ : تعداد مزارع برنج. این مزارع با شماره‌های صفر تا  $R-1$  شماره‌گذاری شده‌اند.
- $L$ : حداکثر مکان ممکن برای انبار و مزارع.
- $X$ : یک آرایه‌ی یک‌بُعدی از اعداد صحیح که از کوچک‌تر به بزرگ‌تر مرتب شده‌است. برای هر  $0 \leq i < R$ ، مزرعه‌ی  $i$ -ام در مکانی  $X[i]$  واقع شده است.
- $B$ : بودجه‌ی ما.

روال شما می‌بایست مکان بهینه‌ی انبار غله را یافته و حداکثر میزان غله‌ای که با بودجه‌ی ما قابل انتقال به انبار هستند را به‌عنوان خروجی بازگرداند.

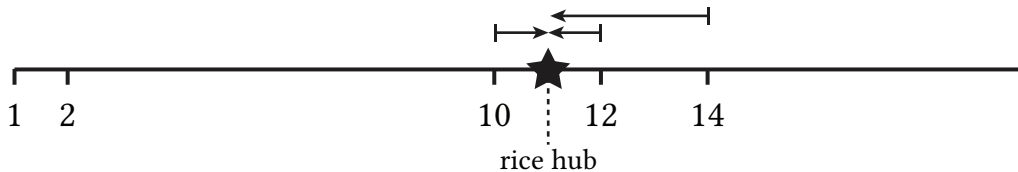
توجه داشته باشید که هزینه‌ی انتقالات ممکن‌ست خیلی بزرگ باشد. بودجه‌ی داده شده در یک عدد صحیح ۶۴-بیتی جا می‌شود و ما به شما توصیه می‌کنیم که در محاسبات خودتان نیز از اعداد صحیح ۶۴-بیتی استفاده کنید. در C/C++ شما می‌توانید از تایپ long long و در پاسکال می‌توانید از تایپ int64 استفاده کنید.

### مثال نمونه

حالتی را در نظر بگیرید که  $R = 5$ ,  $L = 20$ ,  $B = 6$  بوده و  $X$  مشابه لیست کناری باشد.

در این حالت، جواب‌های متعددی برای مکان بهینه‌ی انبارغله وجود دارد: شما می‌توانید این انبار را در هر مکان صحیحی بین مکان ۱۰ الی ۱۴ (شامل این دو مکان) قرار دهید. پس از آن شما قادر خواهید بود تا برنج تولید شده در مزارع موجود در مکان‌های ۱۰، ۱۲ و ۱۴ را به انبار غله انتقال دهید. یکی از این مکان‌های بهینه در شکل زیر نمایش داده شده است.

۱  
۲  
 $X =$  ۱۰  
۱۲  
۱۴



برای هر کدام از این مختصات بهینه، هزینه انتقال حداکثر ۶ بت خواهد شد. به‌وضوح، هیچ مکانی برای انبارغله به ما امکان جمع‌آوری محصول بیش از ۳ مزرعه را نخواهد داد و از همین رو، این جواب بهینه بوده و **besthub** می‌بایست مقدار ۳ را بازگرداند.

### زیرمسئله‌ها

زیرمسئله شماره سه (۲۶ امتیاز)

- $1 \leq R \leq 5,000$
- $1 \leq L \leq 1,000,000$
- $0 \leq B \leq 2,000,000,000$

زیرمسئله شماره یک (۱۷ امتیاز)

- $1 \leq R \leq 100$
- $1 \leq L \leq 100$
- $0 \leq B \leq 10,000$
- هیچ دو مزرعه‌ای در یک مکان یکسان نیستند (فقط در همین زیرمسئله).

زیرمسئله شماره چهار (۳۲ امتیاز)

- $1 \leq R \leq 100,000$
- $1 \leq L \leq 1,000,000,000$
- $0 \leq B \leq 2,000,000,000,000,000$

زیرمسئله شماره دو (۲۵ امتیاز)

- $1 \leq R \leq 500$
- $1 \leq L \leq 10,000$
- $0 \leq B \leq 1,000,000$

## جزئیات پیاده‌سازی

### محدودیت‌ها

- محدودیت زمانی CPU: یک ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- توجه: هیچ محدودیت صریحی روی اندازه‌ی حافظه‌ی پشته (stack) استفاده شده وجود ندارد. میزان حافظه‌ی استفاده شده در پشته به‌عنوان بخشی از حافظه‌ی مصرفی محاسبه می‌شود.

### واسط API

- فولدر پیاده‌سازی: ricehub/
- فایل‌هایی که توسط شرکت کننده می‌بایست پیاده‌سازی شوند: ricehub.c یا ricehub.cpp یا ricehub.pas
- واسط‌های شرکت کننده: ricehub.h یا ricehub.pas
- مصحح نمونه: grader.c یا grader.cpp یا grader.pas
- ورودی مصحح نمونه: grader.in.1, grader.in.2 و ...
- توجه: مصحح نمونه، ورودی را به فرمت زیر می‌خواند:
  - خط اول: مقادیر  $R$ ،  $L$  و در نهایت  $B$ .
  - خط ۲ الی  $R + 1$ : مکان‌های مزارع برنج؛ یعنی در خط  $i + 2$  ام، مقدار  $X[i]$  برای  $0 \leq i < R$  نوشته شده است.
  - خط  $R + 2$  ام: جواب مورد انتظار.
- خروجی مورد انتظار برای ورودی مصحح نمونه: grader.expect.1, grader.expect.2 و ...  
برای این مسئله، هر کدام از این فایل‌ها دقیقاً باید شامل عبارت «**Correct.**» باشند.

## اتومبیل رانی (Race)

هم‌زمان با مسابقه‌ی IOI، پاتایا میزبان مسابقات اتومبیل رانی (IOR)<sup>۱</sup> است. به‌عنوان میزبان، ما باید بهترین مسیر برای مسابقه را پیدا کنیم.

در ایالت پاتایا-چونبوری،  $N$  شهر به‌وسیله‌ی  $N - 1$  بزرگراه به یکدیگر وصل شده‌اند. هر بزرگراه دوطرفه است و دو شهر را به یکدیگر وصل می‌کند و طول آن یک عدد صحیح با واحد کیلومتر است. در ضمن، دقیقاً یک مسیر بین هر دو شهر وجود دارد. یعنی، دقیقاً یک راه برای رفتن از یک شهر به شهر دیگر از طریق دنباله‌ای از بزرگراه‌ها بدون عبور دوباره از یک شهر وجود دارد.

طبق قوانین IOR طول مسیر مسابقه باید دقیقاً  $K$  کیلومتر باشد و شهر شروع و پایانی یکسان نباشند. بدیهی است به منظور جلوگیری از تصادف، هیچ بزرگراهی (و نتیجتاً هیچ شهری) نمی‌تواند دوبار در مسیر ظاهر شود. برای کاهش ترافیک، تعداد بزرگراه‌های مسیر مسابقه باید کمترین باشد.

### وظیفه شما

شما باید روال  $\text{best\_path}(N, K, H, L)$  را بنویسید که پارامترهای زیر را بگیرد:

- $N$ : تعداد شهرها. شهرها از شماره ۰ تا  $N - 1$  شماره گذاری شده‌اند.
- $K$ : طول مسیر مسابقه
- $H$ : یک آرایه دو بعدی که بزرگراه‌ها را نمایش می‌دهد، برای  $0 \leq i < N - 1$  بزرگراه  $i$ -ام دو شهر  $H[i][0]$  و  $H[i][1]$  را به هم وصل می‌کند.
- $L$ : یک آرایه یک بعدی که طول بزرگراه‌ها را نمایش می‌دهد. برای  $0 \leq i < N - 1$  طول بزرگراه  $i$ -ام در  $L[i]$  نگهداری می‌شود.

شما می‌توانید فرض کنید همه مقادیر در آرایه  $H$  بین ۰ و  $N - 1$  (شامل ۰ و  $N - 1$ ) است و بزرگراه‌هایی که در این آرایه نگه داشته شده‌اند تمام شهرها را به هم وصل می‌کنند. شما همچنین می‌توانید فرض کنید که اعداد ذخیره شده در آرایه  $L$  اعداد صحیح بین ۰ تا  $1000000000$  (شامل این دوعدد) هستند.

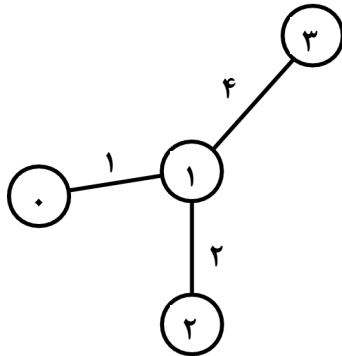
روال شما باید از بین مسیرهای به طول  $K$  کیلومتر، تعداد بزرگراه‌های مسیری که کمترین تعداد بزرگراه را دارد را برگرداند. اگر چنین مسیری وجود نداشته باشد روال شما باید عدد -1 برگرداند.

<sup>1</sup> International Olympiad in Racing

## مثال‌های نمونه

### مثال اول

حالتی را در نظر بگیرید که  $N = 4$  و  $K = 3$  است.



$$L = \begin{matrix} 1 \\ 2 \\ 4 \end{matrix} \qquad H = \begin{matrix} 0 & 1 \\ 1 & 2 \\ 1 & 3 \end{matrix}$$

مسیر مسابقه می تواند از شهر ۰ شروع، از شهر ۱ گذر کرده و در شهر ۲ خاتمه یابد.

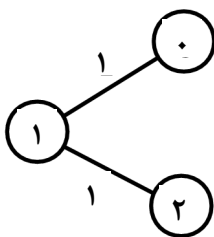
طول این مسیر دقیقاً برابر  $1\text{ km} + 2\text{ km} = 3\text{ km}$  خواهد بود و از دو بزرگراه تشکیل شده است. این بهترین مسیری است که

می توان برای این نمونه متصور شد. در نتیجه روال  $\text{best\_path}(N, K, H, L)$  باید عدد 2 را برگرداند.

### مثال دوم

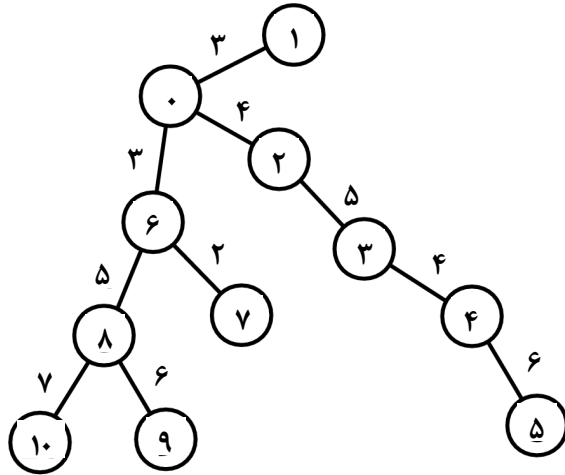
حالتی را در نظر بگیرید که  $N = 3$  و  $K = 3$  است. در این مثال هیچ مسیری به طول  $K$  وجود ندارد. پس روال

$\text{best\_path}(N, K, H, L)$  باید -1 را برگرداند.



$$L = \begin{matrix} 1 \\ 1 \end{matrix} \qquad H = \begin{matrix} 0 & 1 \\ 1 & 2 \end{matrix}$$

### مثال سوم



حالتی را در نظر بگیرید که  $N = 11$  و  $K = 12$  است.

۳	۰ ۱
۴	۰ ۲
۵	۲ ۳
۴	۳ ۴
۶	۴ ۵
$L = ۳$	$H = ۰ ۶$
۲	۶ ۷
۵	۶ ۸
۶	۸ ۹
۷	۸ ۱۰

یک مسیر نمونه شامل سه بزرگراه است: شروع از شهر ۶ و گذر از شهر های ۰ و ۲ و خاتمه در شهر ۳.

یک مسیر دیگر شروع از شهر ۱۰، گذر از شهر ۸ و خاتمه در شهر ۶ می باشد. هر دو مسیر دارای طول ۱۲ کیلومتر هستند

همانطور که در این مثال خواسته شده است. دومین مسیر بهینه است چون مسیری به طول  $K$  شامل فقط یک بزرگراه وجود ندارد.

در نتیجه،  $best\_path(N, K, H, L)$  باید عدد ۲ را برگرداند.

### زیرمسئله‌ها

زیرمسئله شماره یک ( ۹ امتیاز)

$$1 \leq N \leq 100 \quad \bullet$$

$$1 \leq K \leq 100 \quad \bullet$$

• شبکه بزرگراه‌ها، ساده‌ترین خط ممکن را می‌سازد:

برای هر  $0 \leq i < N - 1$ ، بزرگراه  $i$ -ام، شهر  $i$  و

$i + 1$  را به هم وصل می‌کند.

زیرمسئله شماره سه ( ۲۲ امتیاز)

$$1 \leq N \leq 200,000 \quad \bullet$$

$$1 \leq K \leq 100 \quad \bullet$$

زیرمسئله شماره چهار ( ۵۷ امتیاز)

$$1 \leq N \leq 200,000 \quad \bullet$$

$$1 \leq K \leq 1,000,000 \quad \bullet$$

زیرمسئله شماره دو ( ۱۲ امتیاز)

$$1 \leq N \leq 1,000 \quad \bullet$$

$$1 \leq K \leq 1,000,000 \quad \bullet$$

## جزئیات پیاده‌سازی

### محدودیت‌ها

- محدودیت زمانی CPU: سه ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- توجه: هیچ محدودیت صریحی روی اندازه‌ی حافظه‌ی پشته (stack) استفاده شده وجود ندارد. میزان حافظه‌ی استفاده شده در پشته به‌عنوان بخشی از حافظه‌ی مصرفی محاسبه می‌شود.

### واسط API

- فولدر پیاده‌سازی: `race/`
  - فایل‌هایی که توسط شرکت کننده می‌بایست پیاده‌سازی شوند: `race.c` یا `race.cpp` یا `race.pas`
  - واسط‌های شرکت کننده: `race.h` یا `race.pas`
  - واسط مصحح: `race.h` و `racelib.pas`
  - مصحح نمونه: `grader.c` یا `grader.cpp` یا `grader.pas`
  - ورودی مصحح نمونه: `grader.in.1`، `grader.in.2` و ...
  - توجه: مصحح نمونه، ورودی را به فرمت زیر می‌خواند:
    - خط اول: مقادیر  $N$  و  $K$ .
    - خط ۲ الی  $N$  ام: اطلاعات بزرگراه‌ها. یعنی، خط  $i + 2$  ( $0 \leq i < N - 1$ ) شامل  $H[i][0]$ ،  $H[i][1]$  و  $L[i]$  می‌باشد که با یک فاصله جدا می‌شوند.
    - خط  $N + 1$  ام: جواب مورد انتظار.
  - خروجی مورد انتظار برای ورودی مصحح نمونه: `grader.expect.1`، `grader.expect.2` و ...
- برای هر مسئله، هر کدام از این فایلها باید دقیقاً شامل عبارت «**Correct.**» باشد.

## باغ استوایی (Tropical Garden)

Somhed که یک گیاه‌شناس است معمولاً گروه‌های دانش‌آموزی را به یکی از بزرگ‌ترین باغ‌های استوایی تایلند می‌برد. فضای این باغ شامل  $N$  تقاطع (با شماره‌های ۰ تا  $N - 1$ ) و  $M$  راه است. هر راه دو تقاطع متفاوت را مستقیماً به هم وصل می‌کند و می‌توان از هر دو طرف آن را پیمود. هر تقاطع حداقل یک راه متصل به خود دارد. این راه‌ها شامل مجموعه‌های گیاهی زیبایی هستند که Somhed علاقه‌مند به دیدن آن‌هاست. هر گروه مسیر خود را از هر کدام از تقاطع‌ها می‌تواند شروع کنند.

Somhed گیاهان زیبایی گرمسیری را دوست دارد. لذا، از هر تقاطعی او و دانش‌آموزانش راهی را انتخاب می‌کنند که زیباتر از همه است، مگر این‌که هم این زیباترین راه، همین آخرین راهی باشد که آمده‌اند و هم حداقل یک راه دیگر جز این زیباترین راه وجود داشته باشد؛ که در آن صورت، آنها دومین زیباترین راه را انتخاب می‌کنند. واضح است که در صورت نداشتن چاره (هیچ راه دیگری نباشد) آن‌ها از همان راهی که آمده‌اند برمی‌گردند. توجه کنید که چون Somhed یک گیاه‌شناس حرفه‌ایست هیچ دو راهی از نظر او زیبایی یکسانی ندارند.

دانش‌آموزان Somhed علاقه‌ی زیادی به گیاهان ندارند؛ ولی خیلی مایل هستند که ناهار را در رستوران خوبی که در تقاطع شماره  $P$  قرار دارد، بخورند. Somhed می‌داند که دانش‌آموزانش بعد از پیمودن دقیقاً  $K$  راه، گرسنه می‌شوند که  $K$  ممکن است برای هر گروه، عددی متفاوت باشد. Somhed می‌خواهد بداند چند راه متفاوت می‌تواند برای هر گروه انتخاب کند با این شرط که:

- هر گروه می‌تواند از هر تقاطعی کارش را شروع کند
- راه بعدی باید بر اساس قوانین بالا انتخاب شود
- هر گروه باید بعد از پیمودن دقیقاً  $K$  راه در تقاطع شماره  $P$  باشد.

توجه کنید که هر گروه می‌تواند چندین بار از تقاطع شماره  $P$  عبور کند؛ تنها نکته مهم این است که باید در انتها حتماً در تقاطع شماره  $P$  باشند.

### وظیفه شما

با داشتن اطلاعات مربوط به راه‌ها و تقاطع‌ها شما باید جواب را برای هر کدام از  $Q$  گروه پیدا کنید. یعنی برای هر کدام از  $Q$  مقدار  $K$  ی مربوطه را باید بیابید.

روال  $\text{count\_routes}(N, M, P, R, Q, G)$  را بنویسید که پارامترهای زیر را می‌پذیرد

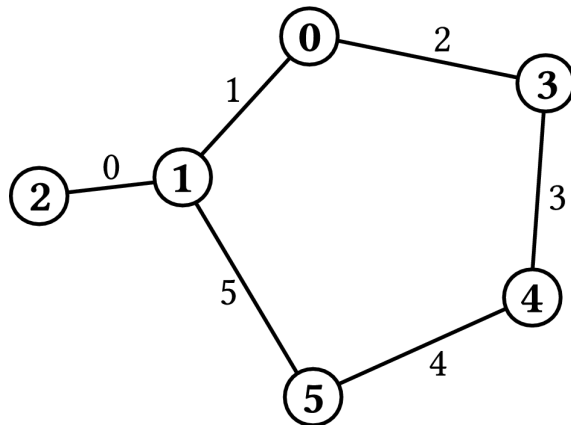
- $N$ : تعداد تقاطع‌ها. تقاطع‌ها با شماره‌های ۰ تا  $N - 1$  مشخص می‌شوند
- $M$ : تعداد راه‌ها. راه‌ها با شماره‌های ۰ تا  $M - 1$  مشخص می‌شوند. راه‌ها به ترتیب نزولی زیبایی داده شده‌اند، یعنی راه  $i$  از راه  $i + 1$  زیباتر است (برای هر مقدار  $0 \leq i < M - 1$ )
- $P$ : تقاطعی که رستوران محبوب در آن قرار دارد.

- $R$ : آرایه‌ای دو بعدی که راه‌ها را نشان می‌دهد. برای هر  $0 \leq i < M$ ، راهی  $i$  تقاطع‌های  $R[i][0]$  و  $R[i][1]$  را به هم وصل می‌کند. به یاد داشته باشید که هر راه، دو تقاطع متفاوت را به هم وصل می‌کند و هیچ دو راهی دو تقاطع یکسان را به هم وصل نمی‌کنند.
  - $Q$ : تعداد گروه‌های دانش‌آموزان.
  - $G$ : یک آرایه‌ی یک بُعدی شامل اعداد  $K$ . برای  $0 \leq i < Q$ ،  $G[i]$  تعداد راه‌هایی است که گروه  $i$  ام می‌پیماید.
- برای  $0 \leq i < Q$ ، برنامه‌ی شما باید تعداد مسیرهای ممکن با دقتاً  $G[i]$  راه را پیدا کند که با پیمودن آن‌ها، گروه  $i$  ام به تقاطع  $P$  می‌رسد. برای هر  $i$ ، برنامه شما باید روال  $\text{answer}(X)$  را برای گزارش این‌که تعداد مسیرها  $X$  است صدا بزند.
- ترتیب جوابها باید مطابق با ترتیب سؤالات باشد. اگر هیچ مسیری وجود نداشت، برنامه شما باید  $\text{answer}(0)$  را فراخوانی کند.

## مثال‌ها

### مثال اول

شکل روبرو را ببینید که در آن  $N = 6, M = 6, P = 0, Q = 1$  و  $G[0] = 3$  است.

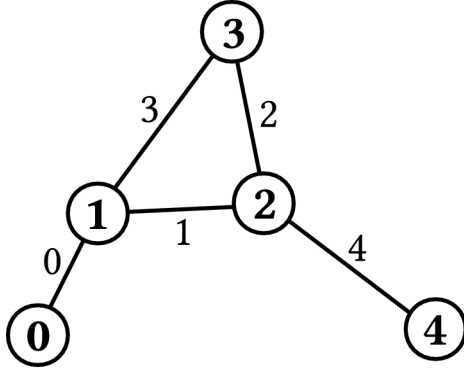


$$R = \begin{matrix} 1 & 2 \\ 0 & 1 \\ 0 & 3 \\ 3 & 4 \\ 4 & 5 \\ 1 & 5 \end{matrix}$$

- توجه کنید که راه‌ها بر اساس ترتیب نزولی زیبایی آمده‌اند به این معنی که راه ۰ زیباترین، راه ۱ دومین زیباترین و ... است. در این مثال فقط دو مسیر به طول ۳ وجود دارد:  $0 \rightarrow 1 \rightarrow 2 \rightarrow 1$  و  $0 \rightarrow 3 \rightarrow 4 \rightarrow 5$ . اولین مسیر از تقاطع ۱ شروع می‌شود. زیباترین راه متصل به آن به تقاطع ۲ ختم می‌شود. گروه، در اینجا چاره‌ای غیر از بازگشت به تقاطع ۱ ندارند. حالا این گروه به جای راه ۰ راه ۱ را انتخاب کرده که آنها را به تقاطع  $P = 0$  می‌برد.
- بنابراین برنامه روال  $\text{answer}(2)$  را فراخوانی می‌کند.

### مثال دوم

شکل زیر را ببینید. در این مثال آن  $N = 5, M = 5, P = 2, Q = 2$  و  $G[0] = 3$  و  $G[1] = 1$  است.



$$R = \begin{matrix} 1 & 0 \\ 1 & 2 \\ 3 & 2 \\ 1 & 3 \\ 4 & 2 \end{matrix}$$

برای گروه اول، فقط یک مسیر هست که طی ۳ مرحله به تقاطع ۲ می‌رسد:

$$1 \rightarrow 0 \rightarrow 1 \rightarrow 2$$

برای گروه دوم، دو مسیر هست که طی یک مرحله به تقاطع ۲ می‌رسند:

$$3 \rightarrow 2 \text{ و } 4 \rightarrow 2$$

لذا در پیاده‌سازی درست `count_routes` باید ابتدا `answer(1)` را فراخوانی کند (برای جواب به گروه اول) و سپس

`answer(2)` را برای جواب به گروه دوم فراخوانی کند.

## زیرمسئله‌ها

زیرمسئله شماره سه (۳۱ امتیاز)

$$2 \leq N \leq 150,000 \quad \bullet$$

$$1 \leq M \leq 150,000 \quad \bullet$$

$$1 \leq Q \leq 2,000 \quad \bullet$$

- هر عنصر آرایه‌ی  $G$  یک عدد صحیح بین ۱ تا  $1,000,000,000$  (شامل خود این دو عدد) است.

زیرمسئله شماره یک (۴۹ امتیاز)

$$2 \leq N \leq 1,000 \quad \bullet$$

$$1 \leq M \leq 10,000 \quad \bullet$$

$$Q = 1 \quad \bullet$$

- هر عنصر آرایه‌ی  $G$  یک عدد صحیح بین ۱ تا ۱۰۰ (شامل خود این دو عدد) است.

زیرمسئله شماره دو (۲۰ امتیاز)

$$2 \leq N \leq 150,000 \quad \bullet$$

$$1 \leq M \leq 150,000 \quad \bullet$$

$$Q = 1 \quad \bullet$$

- هر عنصر آرایه‌ی  $G$  یک عدد صحیح بین ۱ تا  $1,000,000,000$  (شامل خود این دو عدد) است.

## جزئیات پیاده‌سازی

### محدودیت‌ها

- محدودیت زمانی CPU: ۵ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت
- توجه: هیچ محدودیت صریحی روی اندازه‌ی حافظه‌ی پشته (stack) استفاده شده وجود ندارد. میزان حافظه‌ی استفاده شده در پشته به‌عنوان بخشی از حافظه‌ی مصرفی محاسبه می‌شود.

### واسط API

- فولدر پیاده‌سازی: garden/
- فایل‌هایی که توسط شرکت‌کننده می‌بایست پیاده‌سازی شوند: garden.c یا garden.cpp یا garden.pas
- واسط‌های شرکت‌کننده: garden.h یا garden.pas
- واسط مصحح: gardenlib.h یا gardenlib.pas
- مصحح نمونه: grader.c یا grader.cpp یا grader.pas
- ورودی مصحح نمونه: grader.in.1، grader.in.2 و ...
- توجه: مصحح نمونه، ورودی را به فرمت زیر می‌خواند:
  - خط اول: مقادیر  $M$ ،  $N$  و در نهایت  $P$ .
  - خط ۲ الی  $M + 1$  ام: توضیحات راه‌ها؛ یعنی خط  $i + 2$  ام شامل  $R[i][0]$  و  $R[i][1]$  (جدا شده با یک فاصله)، برای  $0 \leq i < M$  است.
  - خط  $M + 2$  ام: مقدار  $Q$ .
  - خط  $M + 3$  ام: آرایه‌ی  $G$  در قالب یک رشته از اعداد صحیح که با فاصله جدا شده‌اند.
  - خط  $M + 4$  ام: آرایه‌ی جواب‌های مورد انتظار، در قالب یک رشته از اعداد صحیح که با فاصله جدا شده‌اند.
- خروجی مورد انتظار برای ورودی مصحح نمونه: grader.expect.1، grader.expect.2 و ...  
برای هر مساله، هر کدام از این فایلها باید دقیقاً شامل عبارت «**Correct.**» باشد.