

هزارتوی افسانه ای

- محدودیت حافظه: ۲۵۶ مگابایت

روزی روزگاری، در سرزمینی دور، یک هزارتوی افسانه‌ای به نام **شبکه طلایی** وجود داشت. گفته می‌شد که در انتهای این هزارتو، گنجی بی‌نظیر در **مربع پایین سمت راست** پنهان شده است. اما این هزارتو طلسمی داشت: هر مربع با یک عدد مشخص شده بود که نشان‌دهنده مقدار **طلایی** بود که مسافر باید برای عبور از آن بپردازد. هدف ساده بود اما چالش‌برانگیز: رسیدن به گنج با **کمترین مقدار طلای ممکن**.

روزی یک تاجر باهوش به نام **پنی‌جو** تصمیم گرفت به این سفر برود. پنی آدم قوی یا سریع نبود، اما ذهنی تیز و حسابگر داشت. پنی با اعتمادبه‌نفس گفت: "این هزارتو رو با فکر و تدبیر شکست می‌دم و طلاهامو حفظ می‌کنم."

قوانین هزارتو روشن بود: پنی فقط می‌توانست به **سمت راست** یا **پایین** حرکت کند. برگشتن به عقب اصلاً ممکن نبود!

ورودی

در خط اول ورودی به شما یک هزارتو $m \times n$ ای در قالب یک رشته داده می‌شود :

$$1 \leq m \leq 200$$

$$1 \leq n \leq 200$$

$$1 \leq maze[i][j] \leq 200$$

خروجی

در خروجی کمترین مقدار طلای ممکن که پنی جو باید خرج کند را چاپ کنید.

مثال

ورودی نمونه ۱

[[1, 3, 1], [1, 5, 1], [4, 2, 1]]

خروجی نمونه ۱

7

توضیح مثال : مسیر بهینه به شکل زیر می شود :

اگر زمین هزارتو را به شکل زیر در نظر بگیریم، کمترین میزان هزینه می شود : $1 \leq 3 \leq 1 \leq 1 \leq 1$ که مجموعاً می شود 7.

1 3 1

1 5 1

4 2 1

ورودی نمونه 2

[[1, 2, 3], [4, 5, 6]]

خروجی نمونه 2

12

مخابره سیگنال

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

تعدادی شهر داریم که می‌خواهیم از شهر اول به باقی شهر ها یک سیگنال بفرستیم.

تعداد شهرها و مسیرها و همچنین زمان مورد نیاز برای عبور سیگنال از هر مسیر را به شما داده خواهد شد و از شما کمترین زمان برای رسیدن سیگنال به همه شهر ها خواسته خواهد شد.

ورودی

در خط اول ورودی تعداد شهر ها n و تعداد مسیر ها m داده خواهد شد. سپس m خط بعدی شامل سه عدد p q k خواهد بود که مسیر از p به q و زمان مورد نیاز برای عبور از مسیر k خواهد بود.

$$2 \leq n \leq 300000$$

$$2 \leq m \leq 600000$$

$$0 \leq k \leq 100$$

خروجی

خروجی باید حداقل زمان مورد نیاز برای رسیدن سیگنال به همه شهر ها باشد.

یا اگر امکان رسیدن به همه شهر ها نبود -1 را برگردانید.

ورودی نمونه ۱

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

خروجی نمونه ۱

2

ورودی نمونه ۲

3 3
2 1 1
1 3 2
2 3 1

خروجی نمونه ۲

-1

Black Friday

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

در یک فروشگاه به مناسبت Black Friday یک طرح خاص در حال اجرا هست. در این فروشگاه n محصول داریم که به ترتیب در یک صف چیده شده اند. در این طرح شما می‌توانید از ابتدا شروع به خریدن محصولات کنید. بعد از خرید محصول i ام شما توانایی این را دارید که i محصول بعد را به صورت مجانی بردارید.

برای مثال اگر شما محصول سوم را بخرید می‌توانید محصول چهارم، پنجم و ششم را به صورت مجانی بردارید. **دقت کنید** که می‌توانید محصول چهارم یا پنجم را بخرید تا محصولات جلوتر را بتوانید به صورت مجانی بردارید.

تعداد و قیمت محصولات به شما داده خواهد شد و از شما کمترین هزینه لازم برای خرید همه محصولات را می‌خواهیم.

ورودی

در خط اول ورودی تعداد محصولات n داده خواهد شد. سپس در خط بعدی قیمت محصولات k داده خواهد شد.

$$1 \leq n \leq 20000$$

$$1 \leq k \leq 1000000$$

خروجی

شما باید حداقل هزینه لازم برای خرید همه محصولات را محاسبه کرده و خروجی دهید .

ورودی نمونه ۱

3

3 1 2

میوه اول را با قیمت 3 خریداری کنید. میوه دوم را با قیمت 1 بخرید. میوه سوم را رایگان بگیرید.

خروجی نمونه ۱

4

ورودی نمونه ۲

4

1 10 1 1

میوه اول را با قیمت 1 خریداری کنید. میوه دوم را به صورت رایگان بگیرید. میوه سوم را به قیمت 1 بخرید.
میوه چهارم را به صورت رایگان بگیرید.

خروجی نمونه ۲

2

مسابقات سازه ماکارونی

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: 256 مگابایت

هفته آینده قرار است که یک مسابقه ی سازه ماکارونی در دانشگاه برگزار شود. تیم دانشگاه ما در آزمایشگاه خود یک سری نمونه ساخته است که در برابر حداکثر میزان وزن های متفاوتی مقاوم هستند. آن ها از هر الگو به مقداری محدود نمونه ساخته اند و می خواهند که در برابر وزن های مختلف میزان مقاومت آن ها را تست کنند اما نمی خواهند که همه ی نمونه های خود را برای آزمایش ها از دست بدهند. فرض کنید آن ها هر بار می خواهند مقاومت یک سازه را بسنجند. در آزمایشگاه برای هر تست، n سطح وزنه در نظر گرفته شده است. در هر تست نیز k نمونه داریم که می توانیم برای آزمایش از آن استفاده کنیم. با گذاشتن هر سطح وزنه ای روی سازه، سازه یا می شکند یا مقاومت می کند و سالم می ماند و می توانیم دوباره از آن سازه برای آزمایش استفاده کنیم. یک سطح وزنه w وجود دارد که هر میزان وزن بالاتری از آن روی سازه قرار دهیم منجر به شکستن سازه خواهد شد و هر میزان وزنی به اندازه ی w یا کمتر از آن آسیبی به سازه وارد نخواهد کرد. می خواهیم با تعداد محدود سازه ای که داریم با کمترین تعداد آزمایش، مقدار w را متوجه بشویم.

ورودی

در خط اول ورودی تعداد سازه هایی که می توانیم برای هر آزمایش استفاده کنیم k و تعداد سطح وزنه ها n به شما داده می شود.

$$1 \leq k \leq 100$$

$$1 \leq n \leq 1000000$$

خروجی

در خروجی کمترین تعداد باری که نیاز هست آزمایش تکرار شود با توجه به سطح وزنه ها و سازه هایی که داریم تا متوجه سطح تحمل سازه w شویم، را چاپ کنید. دقت کنید که می خواهیم مطمئن شویم که سطح

w واقعا سطح تحمل سازه است پس باید کمترین تعداد تکرار آزمایش را طوری بیابید که مطمئن باشید با شروع از یک سطح وزنه در بدترین حالت نیز مقدار w را پیدا می کنید.

$$0 \leq w \leq n$$

مثال

ورودی نمونه ۱

1 2

خروجی نمونه ۱

2

توضیح مثال : فرض کنید یک سازه برای آزمایش در اختیار دارید و دو سطح وزنه دارید. می خواهید بدانید سطح تحمل سازه شما در کدام یک از سطوح 0، 1، 2 قرار دارد. وزنه سطح 1 را روی سازه قرار می دهیم. اگر بشکند پس می توانیم نتیجه بگیریم که سطح تحمل سازه 0 است. اگر نشکند وزنه سطح 2 را روی سازه قرار می دهیم. اگر بشکند یعنی سطح تحمل سازه 1 است و اگر نشکند یعنی سطح تحمل سازه 2 است. پس مینیمم تکرار آزمایش برای اینکه مطمئن بشویم سطح سازه چقدر است برابر است با 2.

ورودی نمونه 2

2 6

خروجی نمونه 2

3

توضیح مثال : فرض کنید دو سازه برای آزمایش در اختیار دارید و 6 سطح وزنه دارید. می خواهید بدانید سطح تحمل سازه شما در کدام یک از سطوح 0، 1، 2، 3، 4، 5، 6 قرار دارد. یک راه حل این است که وزنه سطح 4 را

روی سازه قرار دهیم. اگر سازه بشکند یعنی ممکن است هر یک از سطوح 0,1,2,3 سطح تحمل باشند. اگر سازه نشکند یعنی ممکن است هر یک از سطوح 5 یا 6 سطح تحمل سازه باشند. در حالت اول نیاز به حداقل دو بار تکرار دیگر آزمایش با تنها سازه ای که در اختیارتان هست دارید. در حالت دوم نیز در بدترین حالت باید دو بار آزمایش را تکرار کنید تا به میزان w مطمئن برسید. پس در مجموع به 3 بار تکرار آزمایش نیاز دارید.

ورودی نمونه 3

3 14

خروجی نمونه 3

4

جستجوگر بهینه

- محدودیت زمان: 5 ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

یک شرکت بهینه‌سازی جستجو برای پایگاه‌های داده متنی وجود دارد که به تحلیل و تبدیل داده‌های متنی در سیستم‌های جستجو می‌پردازد. این شرکت دو رشته متنی اولیه و نهایی دارد که هر دو شامل کاراکترهای حروف کوچک انگلیسی بوده و طول یکسان (n) دارند.

علاوه بر این، این شرکت از یک مدل تبدیل پیشرفته استفاده می‌کند که اجازه می‌دهد تا برخی از کاراکترهای موجود در رشته اولیه به کاراکترهای جدید در رشته نهایی تبدیل شوند. این مدل از دو آرایه A و B به همراه یک آرایه عددی C استفاده می‌کند که در آن هر ایندکس از آرایه عددی C نشان‌دهنده هزینه تبدیل کاراکتر آن ایندکس در آرایه A به کاراکتر متناظر آن ایندکس در آرایه B می‌باشد. طول این سه آرایه برابر (m) می‌باشد.

در هر عملیات جستجو و تبدیل داده، سیستم قادر است تا یک کاراکتر را از رشته اولیه انتخاب کرده، آن را به کاراکتر متفاوت دیگری در رشته نهایی و با صرف هزینه ای مشخص تبدیل کند.

هدف شرکت این است که حداقل هزینه تبدیل رشته اولیه به رشته نهایی را محاسبه کند. اگر تبدیل رشته ابتدایی به رشته نهایی ممکن نباشد، مقدار -1 بازگردانده خواهد شد.

محدودیت‌ها:

- رشته‌های اولیه، نهایی و آرایه های A و B شامل حروف کوچک انگلیسی هستند.

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

$$1 \leq m \leq 2000$$

- هر ایندکس از آرایه C دارای حداقل 1 و حداکثر 10^6 مقدار است.
- هیچ کاراکتری مجاز به تبدیل شدن به خودش نیست!

نکته: نام گذاری آرایه ها صرفاً برای جلوگیری از ایجاد ابهام می باشد.

ورودی نمونه ۱

4 6
 abcd
 acbe
 a b c c e d
 b c b e b e
 2 5 5 1 2 20

خروجی نمونه ۱

28

برای تبدیل رشته abcd به acbe، مراحل زیر انجام می‌شود:

- تغییر کاراکتر در ایندکس 1 از b به c با هزینه 5.
- تغییر کاراکتر در ایندکس 2 از c به e با هزینه 1.
- تغییر کاراکتر در ایندکس 2 از e به b با هزینه 2.
- تغییر کاراکتر در ایندکس 3 از d به e با هزینه 20.

جمع هزینه‌ها برابر با 28 است.

ورودی نمونه ۲

10 2
 tqwqscktnp
 omhrngpigl
 l k
 s e
 5 6

خروجی نمونه ۲

-1

این امکان وجود ندارد که رشته اولیه را به رشته نهایی مشخص شده تبدیل کنیم.

عوارض جاده‌ای (امتیازی)

- محدودیت زمان: ۳ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

در یک منطقه وسیع و سرسبز که شامل چندین روستا است، شما مسئول طراحی شبکه جاده‌ای برای دسترسی راحت‌تر به این روستاها هستید. هر روستا به دیگر روستاها از طریق جاده‌های مختلفی متصل است و شما تصمیم دارید که برای هر روستا، یک شبکه جاده‌ای بهینه طراحی کنید که هزینه‌اش به حداقل برسد.

این هزینه شامل عوارض جاده‌ای است که برای تعمیر و نگهداری جاده‌ها باید پرداخت شود. شما قصد دارید برای هر روستا محاسبه کنید که از طریق کدام جاده‌ها می‌تواند به سایر روستاها دسترسی پیدا کند و هزینه عوارض جاده‌ها نباید از حد معینی بیشتر شود. به این ترتیب، شما می‌توانید برای هر روستا یک شبکه جاده‌ای بهینه بسازید که کمترین هزینه عوارض را داشته باشد.

ورودی‌ها:

۱. تعداد روستاها (n) که از 0 تا $n-1$ شماره‌گذاری شده‌اند.
۲. مسیرهای موجود به تعداد m شامل اطلاعات جاده‌ها و هزینه‌های عوارضی آن‌ها است. هر جاده بین دو روستا $from_i$ و to_i با هزینه Fee_i مشخص می‌شود.
۳. یک مقدار $maxFee$ که نشان می‌دهد حداکثر هزینه‌ای که برای عبور از جاده‌ها و رسیدن به یک روستا می‌توانید بپردازید چقدر است.

هدف:

- برای هر روستا باید محاسبه کنید که چقدر از دیگر روستاها را می‌تواند با هزینه کمتر از یا برابر با $maxFee$ از طریق جاده‌های مختلفی که در اختیار دارد، به هم متصل کند.
- در نهایت، شما باید روستایی را پیدا کنید که کمترین تعداد روستاهایی را داشته باشد که از طریق جاده‌هایش به آن‌ها دسترسی دارد و هزینه عوارض جاده‌ها از $maxFee$ تجاوز نمی‌کند.
- اگر چندین روستا با تعداد همسایگان یکسان وجود داشته باشد، باید روستای با شماره بزرگتری را انتخاب کنید.

ورودی نمونه ۱

4 4
 0 1 3
 1 2 1
 1 3 4
 2 3 1
 4

خروجی نمونه ۱

3

شهرهای همسایه در یک آستانه ($\text{maxFee} = 4$) برای هر شهر به شرح زیر هستند:

- شهر 0 --> [شهر 1، شهر 2]
- شهر 1 --> [شهر 0، شهر 2، شهر 3]
- شهر 2 --> [شهر 0، شهر 1، شهر 3]
- شهر 3 --> [شهر 1، شهر 2]

شهرهای 0 و 3 هر کدام دو شهر همسایه دارند که در فاصله‌ای کمتر یا مساوی با ۴ از آنها قرار دارند، اما باید شهر 3 را بازگردانیم چون از نظر عددی بزرگتر است.

ورودی نمونه ۲

5 6
 0 1 2
 0 4 8
 1 2 3
 1 4 2
 2 3 1
 3 4 1
 2

خروجی نمونه ۲

0

- شهر 0 <-- [شهر 1]
- شهر 1 <-- [شهر 0، شهر 4]
- شهر 2 <-- [شهر 3، شهر 4]
- شهر 3 <-- [شهر 2، شهر 4]
- شهر 4 <-- [شهر 1، شهر 2، شهر 3]

شهر 0 فقط 1 شهر همسایه دارد که در فاصله کمتر یا مساوی با ۲ از آن قرار دارد، بنابراین باید شهر 0 را بازگردانیم.