

رنگ آمیزی پرانتزی

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

یونس روی میز، تکه کاغذی پیدا کرده است که روی آن یک رشته پرانتزگذاری معتبر به صورت رنگی نوشته شده است و هر حرف آن به رنگ خاصی است. او پی برد که یکی از دوستانش با سلیقه‌ی خاصی پرانتزگذاری را به صورت رنگی برای به دست آوردن دل او نوشته است. او متوجه می‌شود که دوستش از سبک رنگ آمیزی کوبیسم استفاده کرده است.

رشته‌ی پرانتزگذاری s با طول n را در نظر بگیرید. برای هر حرف i ، m_i را برابر با اندیس پرانتز باز یا بسته متناظر با حرف i رشته در نظر بگیرید. از آنجا که این پرانتزگذاری معتبر است، مقدار m_i به ازای هر i وجود دارد. برای مثال اگر دنباله پرانتزگذاری ما $((()))$ باشد، دنباله‌ی m برابر با $\{4, 3, 2, 1, 6, 5\}$ خواهد بود. این رشته یک رنگ آمیزی کوبی است اگر ویژگی‌های زیر را داشته باشد:

۱. هر حرف به یکی از رنگ‌های ۱ تا k رنگ شده باشد.

۲. رنگ حرف i و رنگ حرف m_i باید برابر باشد.

۳. اگر $m_i \neq i - 1$ رنگ حرف i با $i - 1$ متفاوت باشد.

۴. اگر $m_i \neq i + 1$ رنگ حرف i با $i + 1$ متفاوت باشد.

یونس می‌خواهد تعداد رنگ آمیزی‌های کوبی متفاوت s با k رنگ را زیر آن تکه کاغذ بنویسد تا مصطفی را بیشتر حیرت زده کند! از آنجا که این عدد خیلی بزرگ است، باقی‌مانده آن را بر $10^9 + 7$ برایش به دست آورید.

ورودی

در سطر اول ورودی دو عدد n و k آمده است. سپس در سطر بعد رشته پرانتزگذاری s آمده است. تضمین می‌شود s یک پرانتزگذاری معتبر است.

$$1 \leq k \leq n \leq 200\,000$$

$$|s| = n$$

خروجی

در تنها سطر خروجی باقی‌مانده، تعداد روش‌های رنگ‌آمیزی کوبی s با k رنگ را بر $10^9 + 7$ چاپ کنید.

نمونه

ورودی ۱

```
4 2
(( ))
```

خروجی ۱

```
2
```

ورودی ۲

```
6 3
(( ))( )
```

خروجی ۲

```
12
```

گرافیت

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

یک گراف ساده به نام G با n راس و m یال داریم. رأس‌های این گراف با اعداد 1 تا n شماره‌گذاری شده‌اند. در هر عملیات می‌توانیم یکی از دو عدد صحیح و مثبت مثل u و v را انتخاب کنیم به‌طوری‌که $1 \leq u \neq v \leq n$ و یکی از دو عملیات زیر را انجام دهیم اگر یال uv در G موجود است، آن را حذف کنیم. اگر یال uv در G موجود نیست، آن را به G اضافه کنیم.

می‌خواهیم با این عملیات‌ها G را به یک درخت، تبدیل کنیم. به شما گراف G داده می‌شود و از شما می‌خواهیم کمترین تعداد عملیات لازم برای تبدیل G به یک درخت را محاسبه کنید.

ورودی

در سطر اول ورودی، دو عدد صحیح n و m که با یک فاصله از هم جدا شده‌اند، داده می‌شود.

$$1 \leq n \leq 100\,000$$

$$0 \leq m \leq 500\,000$$

در m سطر بعدی، در هر سطر، دو عدد صحیح u و v که با یک فاصله از هم جدا شده‌اند، داده می‌شود. uv یک یال از G است.

$$1 \leq u < v \leq n$$

خروجی

کمترین تعداد عملیات لازم برای تبدیل G به یک درخت.

نمونه

ورودی ۱

5 4
1 2
1 3
2 3
4 5

خروجی ۱

2

ورودی ۲

3 2
1 2
1 3

خروجی ۲

0

جوادفشاری

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

آقا جواد که فردی بسیار پولپرست است، جهت صرفه‌جویی در تعداد حروف پیامک‌هایش به فشرده‌سازی رشته‌ها روی آورده.

روش آقا جواد برای فشرده‌سازی به این صورت است:

او در ابتدا جایگشتی از اعداد 1 تا k را انتخاب می‌کند. سپس رشته‌ی خود را به دسته‌های پشت سر هم k حرفی تقسیم می‌کند (طول رشته باید به k بخش‌پذیر باشد) و روی هر دسته از حروف، جایگشت خود را اعمال می‌کند. سپس رشته‌ی به‌دست‌آمده را به‌وسیله‌ی روش RLE که در ادامه توضیح داده خواهد شد، فشرده می‌کند.

اعمال جایگشت p روی یک دسته از k حرف، یعنی حرف p_1 را در جایگاه اول قرار داده، حرف p_2 را در جایگاه دوم و ... برای مثال اعمال جایگشت $\{3, 1, 4, 2\}$ روی رشته‌ی $abcd$ آن را تبدیل به $cadb$ می‌کند. اعمال این جایگشت با دسته‌دسته کردن روی رشته‌ی $abcdefgh$ ، رشته‌ی $cadbgehf$ را نتیجه می‌دهد.

رشته‌ی جایگشت‌داده‌شده توسط RLE (یا Run-Length Encoding) فشرده می‌شود. جهت جلوگیری از محاسبات پیچیده، طول رشته‌ی فشرده‌شده را برابر تعداد گروه حرف‌های برابر پشت سر هم در نظر می‌گیریم. برای مثال طول رشته‌ی $aabcaa$ پس از فشرده‌شدن توسط RLE را ۴ در نظر می‌گیریم (یک گروه دوحرفی a ، یک گروه تک‌حرفی b ، یک گروه تک‌حرفی c و یک گروه دوحرفی a).

آقا جواد می‌خواهد پیامکی را با روش گفته‌شده فشرده کرده و بفرستد. واضح است که طول رشته‌ی نهایی به جایگشت انتخاب‌شده مربوط است. شما با داشتن متن پیامک آقا جواد، جایگشتی انتخاب کنید که پس از فشرده‌سازی به‌وسیله‌ی آن طول پیامک کمینه شود و این طول کمینه را خروجی دهید.

ورودی

سطر اول تنها شامل عدد k است.

در سطر بعدی پیامک آقا جواد به‌صورت یک رشته از حروف کوچک انگلیسی به طول حداکثر ۵۰۰۰۰ آمده‌است.

$$2 \leq k \leq 16$$

خروجی

در تنها سطر خروجی یک عدد چاپ کنید که برابر کمترین طول ممکن برای پیامک آقا جواد است.

نمونه

ورودی ۱

```
4
abcabcabcabc
```

خروجی ۱

```
7
```

در این مثال با انتخاب جایگشت $\{1, 4, 3, 2\}$ نتیجه‌ی بهینه به دست می‌آید.

پاستا پنه آلفردو

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

در یک رستوران k آشپز کار می‌کنند. n پاستا پنه آلفردو باید آماده شود. هر آماده‌سازی ۳ مرحله دارد و برخی مراحل آماده‌سازی می‌تواند هم‌زمان انجام شود. برای حاضر کردن i امین آن‌ها باید ابتدا یکی از آشپزها a_i دقیقه پاستای آن را حاضر کند و روی میز ۱ بگذارد. سپس یکی از آشپزها پاستای آن را از روی میز ۱ بردارد و در b_i دقیقه پاستا پنه‌ی آن را حاضر کند و روی میز ۲ بگذارد. سپس یکی از آشپزها پاستا پنه‌ی آن را از روی میز ۲ بردارد و در c_i دقیقه پاستا پنه آلفردو را حاضر کند و تحویل مشتری دهد. یک محدودیت مهم، این است که روی میز ۱ حداکثر x پاستا و روی میز ۲ حداکثر y پاستا پنه جا می‌شود. برنامه‌ای برای کار کردن آشپزها ارائه دهید که زمان آماده‌سازی این n پاستا پنه آلفردو به وسیله‌ی k آشپز رستوران کمینه شود.

آشپزها با اندیس‌های ۱ تا k شماره‌گذاری شده‌اند.

مدت زمان برداشتن یا گذاشتن پاستاها روی میز و تحویل دادن آن به مشتری ناچیز است و می‌توان آن را ۰ در نظر گرفت.

دقت کنید که بعد از این‌که آشپزی آماده‌سازی را انجام داد، باید پاستا را روی میز بگذارد؛ یعنی اگر در زمان t آشپزی پاستا را حاضر کرد، آن را روی میز می‌گذارد و ممکن است آشپز دیگری در همان لحظه پاستا را از روی میز بردارد ولی نباید در زمان t بیشتر از حد مجاز پاستا روی میز باشد. به عبارت دیگر، در زمان t ، پاستا روی میز است مستقل از این‌که آشپز دیگری آن را در لحظه t بردارد.

اگر آشپزی در زمان t پاستا را بردارد و آماده‌سازی آن m دقیقه طول بکشد، دقیقاً در زمان $t + m$ باید پاستا را روی میز بگذارد و نمی‌تواند دیرتر یا زودتر آن را روی میز بگذارد.

آشپزها تنها در زمان‌هایی که حسابی هستند، می‌توانند شروع به انجام کاری کنند؛ یعنی زمان ۰، ۱، ۲، ۳ و ...

دقت کنید که هرگاه یک آشپز درگیر یک آماده‌سازی بشود (مثلاً آماده کردن پاستا، آماده کردن پاستا پنه آلفردو از روی پاستا پنه)، باید آن آماده‌سازی را به‌طور کامل انجام دهد و سپس سراغ کار بعدی خود برود. همچنین هر آشپز در هر لحظه یک آماده‌سازی می‌تواند انجام دهد.

در این سؤال، هرچه برنامه‌ی شما بهینه‌تر باشد و در زمان کم‌تری همه‌ی پاستا پنه آلفردوها حاضر شود، نمره‌ی بیشتری دریافت می‌کنید. به عبارت دیگر نیازی نیست در کم‌ترین زمان ممکن همه‌ی پاستا پنه آلفردوها را آماده کنید و نمره‌ی شما از این سؤال صرفاً به اختلاف زمان آماده کردنی که شما به دست می‌آورید و جواب کمینه‌ی مسئله بستگی دارد.

ورودی

در خط اول ورودی، به‌ترتیب چهار عدد n (تعداد پاستاها)، k (تعداد آشپزها)، x (تعداد پاستایی که روی میز ۱ جا می‌شود)، y (تعداد پاستا پنه‌ای که روی میز ۲ جا می‌شود) آمده‌است. سپس در n خط بعدی، در هر خط ۳ عدد a_i ، b_i و c_i آمده است که به ترتیب زمان موردنیاز برای آماده کردن پاستا، پاستا پنه و پاستا پنه آلفردو i می‌باشد.

$$1 \leq n, x, y, k \leq 1\,000$$

$$1 \leq a_i, b_i, c_i \leq 1\,000\,000$$

خروجی

خروجی شما ترتیب آماده‌سازی‌ها است که باید شامل $3n$ خط باشد که در خط i م شما باید سه عدد s_i ، t_i و f_i خروجی دهید که به این معناست که در دقیقه‌ی t_i آشپز s_i به آماده‌سازی f_i مین غذا می‌پردازد (اگر پاستای آن حاضر بود، آن را به پاستا پنه تبدیل می‌کند و اگر پاستا پنه‌ی آن حاضر بود، آن را به پاستا پنه آلفردو تبدیل می‌کند، وگرنه پاستای آن را حاضر می‌کند).

خروجی شما باید معتبر باشد؛ یعنی شرطهای زیر برای آن برقرار باشد:

$$0 \leq t_i \leq 3 \times 10^9$$

$$1 \leq s_i \leq k$$

$$1 \leq f_i \leq 1\,000$$

در لحظه t_i پاستا f_i در حال آماده‌سازی نباشد و آشپز مشغول آماده‌سازی پاستا دیگری نباشد. ممکن است در لحظه t_i کار آشپز تمام شود و کار دیگری را شروع کند و یا در آن لحظه پاستا به مرحله‌ی بعدی آماده‌سازی برود و آشپز پاستا را در همان لحظه از روی میز بردارد.

نباید پاستایی را بیشتر از ۳ بار در خروجی چاپ کنید.

هیچ‌گاه نباید روی میزی، بیشتر از حد مجازش پاستا باشد.

برای هر $i \leq 2$ داریم $t_{i-1} \leq t_i$.

برای هر تست، در صورتی‌که خروجی شما معتبر نباشد، نمره‌ی آن تست را نمی‌گیرید.

نمونه

ورودی ۱

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

خروجی ۱

```
0 1 1
0 2 2
1 1 1
2 2 2
3 2 3
6 1 3
6 2 2
7 2 1
8 1 3
```

زمانی که طول می‌کشد تا همه‌ی پاستا پنه آلفردوها حاضر شوند، ۱۳ دقیقه است و زودتر از این زمان نمی‌شود تمام پاستا پنه آلفردوها را حاضر کرد.

