Palindromiczny Żabuś (A)

Limit pamięci: 64 MB   Limit czasu: 1.00 s

-... Albert, Salami, Śmieszek, Nerwus, Tom, Tomasz, Tamburyn, Głowonóg, McGula, Karczoszek, Pingwin, Pete i Steve. Ale chyba najgorsze imię dla tego żabusia to...

-Zaraz, czekaj, Greg – przerwał bratu Wirt, rozglądając się niespokojnie wśród otaczających ich drzew. – Gdzie... my jesteśmy? – Nagle chwycił się za głowę, rwąc włosy z rozpaczy. Krążąc nerwowo po leśnej ściółce, zaczął wygłaszać dramatyczny monolog: – Choć zabłądziłem, moje zranione serce zostało w domu i złamane spoczywa na cmentarzu utraconej miłości...

Jednak Greg miał na głowie sprawę znacznie ważniejszą niż zabłądzenie w lesie czy problemy miłosne starszego brata. W końcu kto, jeśli nie on, miał wybrać imię dla żabusia?

Greg chciał, aby imię żabusia było *prawie-palindromem*. Co więcej, z racji jego młodego wieku, zdarzało mu się czasem mylić litery. Na przykład nie potrafił odróżnić od siebie liter S i Z, a do tego jego pamięć bywała zawodna – jeśli zapomniał, jaką literę miał wpisać, zastępował ją symbolem $*$, który mógł oznaczać dowolną literę.

Dane jest **słowo długości** N oraz k **- ograniczenie na liczbę błędów**.

Słowo zbudowane jest z **wielkich liter alfabetu łacińskiego oraz z symbolu** $*$**, który może reprezentować dowolną wielką literę** $A - Z$.

Greg uzna, że imię dla żabki jest dobre, jeśli, jest **palindromem z nie więcej niż** k **błędami**. Dla przypomnienia:

1. Słowo jest *palindromem*, jeśli czytane od lewej do prawej jest takie samo, jak czytane od prawej do lewej (na przykład słowo $KAJAK$).
2. Słowo jest *palindromem z* k *błędami*, jeśli po zamianie k liter, staje się ono palindromem (na przykład słowo $KAJMAK$ jest *palindromem z* 1 *błędem* - można zamienić literę J na M, aby uzyskać palindrom).
3. Słowo jest *palindromem z nie więcej niż* k *błędami*, jeśli jest palindromem z l błędami, dla $l \leq k$ (słowa $KAJMAK$ i $KAJAK$ są *palindromami z nie więcej niż 1 błędem*, ale słowo $BOJKOT$ nie jest).

Co więcej, **litera** S **i** Z **to według niego ta sama litera**, więc w słowie $ZAW * AS$ nie widzi żadnego błędu (wyjaśnienie: myląc litery S i Z dopasowuje do siebie pierwszą i ostatnią literę, następnie dopasowuje do siebie literę A z drugiej i piątej pozycji, Litera W z trzeciej pozycji pasuje do * z czwartej pozycji w słowie).

Wejście

W pierwszym wierszu wejścia znajduje się liczba N oznaczająca długość słowa, oraz liczba k - maksymalna liczba błędów.

W drugim wierszu wejścia znajduje się słowo długości N złożone z wielkich liter alfabetu łacińskiego oraz z symbolu $*$.

Wyjście

W jedynym wierszu wyjścia znajduje się odpowiedź na pytanie - *czy imię podane na wejściu spełnia warunki Grega?*.

TAK - jeśli spełnia warunki, NIE w przeciwnym przypadku.

Ograniczenia

$1 \leq N \leq 1000$,

$0 \leq k \leq N/2$.

Przykład

Wejście	Wyjście	Wyjaśnienie
`5 0` `ZAB*S`	`TAK`	Według Grega podane słowo jest palidromem, nie zawiera żadnego błedu. Pierwsza litera S pasuje do ostatniej litery Z, druga litera A pasuje do *, natomiast środkowa litera psuje do samej siebie.

Wejście	Wyjście	Wyjaśnienie
`5 0` `ZABUS`	`NIE`	To słowo nie spełnia podanych warunków - druga litera A nie pasuje do litery U, więc liczba błędów to 1, co przekracza limit 0 błędów, który był podany na wejściu.

Wejście	Wyjście	Wyjaśnienie
`10 3` `G*REGSFROG`	`TAK`	To słowo Greg uznaje za palindrom. Pierwsza i ostatnia litera to G, następnie druga litera $*$ pasuje do drugiej od końca litery O, trzecia i trzecia od końca litera to R, następnie występują dwa błedy - litera E nie pasuje do F, dalej G nie pasuje do S.

Wejście	Wyjście	Wyjaśnienie
`11 1` `KARCZOS*ZEK`	`NIE`	W tym słowie Greg widzi dwa błędy. Limit błędów jest przekroczony o 1.

Sopelkowo (B)

Limit pamięci: 64 MB   Limit czasu: 1.00 s

Sopelkowo, Lodowa Kraina.

Arktos, kręcąc się i wirując w swojej lodowej komnacie, myśląc o swoich nowych, lśniących łyżwach, wpadł na kolejny genialny pomysł:

– Jakubie, zrób mi lodowisko! Chcę mieć arenę lodową przed pałacem, na której będę mógł trenować łyżwiarstwo figurowe i zadziwiać moich poddanych piruetami.

Jakub, wierny i oddany sługa Arktosa, zna swojego króla jak nikt inny.

– Co za bałwan… – mruczy pod nosem. – Co on może wiedzieć o łyżwiarstwie? I czy on ma w ogóle nogi, na które nałoży te swoje nowe łyżwy?

Poprawił monokl i spojrzał na stare zdjęcie z dzieciństwa, na którym, jeszcze jako młody pingwinek, wykonuje perfekcyjnego axla. Łezka zakręciła mu się w oku i spadła na posadzkę jako maleńki sopelek. Choć uważał pomysł króla za absurdalny, nie miał zamiaru odmówić. Jakub wiedział, że sprzeciwianie się Arktosowi mogłoby zakończyć się w najlepszym wypadku na lodowym wygnaniu.

Problem był jednak poważny – teren przed pałacem Arktosa, zwanym Mroźnogrodem, był pełen gór i pagórków. Aby stworzyć idealnie płaskie lodowisko, Jakub musi wyrównać wysokości wszystkich górek. Każda zmiana wysokości kosztuje, a Jakub wie, że król Arktos nie przepada za niepotrzebnymi wydatkami (w końcu wydaje wszystko na nowe gadżety do lodowego pałacu).

Dokładniej, **wyrównując górę o wysokości h do poziomu X, Jakub zapłaci** $(X-h)^2$.

Teraz Jakub zwraca się do Ciebie z prośbą o pomoc. Twoim zadaniem jest obliczyć **minimalny koszt** wyrównania wszystkich górek tak, aby powstała **idealnie płaska** tafla lodu.

Wejście

W pierwszym wierszu wejścia znajduje się liczba N ($1 \le N \le 10^6$), która oznacza liczbę górek przed pałacem Arktosa.

W drugiej linii wejścia znajduje się ciąg liczb całkowitych $h_1, h_2, \ldots, h_N$ ($0 \le h_i \le 10^6$) - ich wysokości.

Wyjście

W jedynym wierszu wyjścia powinny jedna liczba, określająca **minimalny koszt** wyrównania gór. Wysokość do której wyrówane zostaną góry jest liczbą całkowitą.

Ograniczenia

$1 \le N \le 10^6$,

$0 \le h_i \le 10^6$.

Przykład

Wejście
```
5
1 1 1 1 2
```

Wyjście
```
1
```

Wyjaśnienie
Jakub wyrównuje góry do poziomu 1, płaci za to $(1-1)^2 + (1-1)^2 + (1-1)^2 + (1-1)^2 + (1-2)^2 = 1$

Wejście
```
6
0 2 3 1 4 6
```

Wyjście
```
24
```

Wyjaśnienie
Koszt wyrównania gór do poziomu 3 to $(3-0)^2 + (3-2)^2 + (3-3)^2 + (3-1)^2 + (3-4)^2 + (3-6)^2 = 24$

Lentilky (C)

Limit pamięci: 64 MB Limit czasu: 1.00 s

Nikogo ani niczego nie było, tylko ten talerz z kaszą.

Po wielkiej uczcie z kaszy Żwirek i Muchomorek wciąż czuli się najedzeni, ale jak zwykle szukali nowych przygód. Tym razem na miękkim mchu leśnej polanki znaleźli małe, kolorowe cukiereczki rozsypane jak drobne skarby.

– To chyba magiczne kamyki z leśnej wróżby! – zastanawiał się Żwirek, biorąc jeden z cukierków w dłoń. – Magiczne? To pewnie cukrowe perły! – odparł Muchomorek, który już próbował jednego z nich.

Oczy Muchomorka rozświeciły się – to nie żadne kamyki ani perły, tylko Lentilky, ich ulubione słodkości!

– Stop! – powiedział Żwirek, podnosząc rękę. – Po wczorajszej uczcie nie możemy ich jeść bez umiaru. Mam pomysł: zagramy w grę!

Ułożyli Lentilky w dwóch kolorach - czerwonym i niebieskim na stosie (choć dla widzów przed telewizorem różniły się jedynie odcieniem szarości). Ustalili zasady:
1. Grają na zmianę, każdy z nich w swoim ruchu może zdjąć **od 1 do k cukierków z góry stosu**.
2. **Wygra** ten, kto zdejmie **ostatni czerwony cukierek**.

Obydwaj wyciągnęli się wygodnie na mchu, zmrużyli oczy i wytężyli umysły. Opracowali optymalne strategie i rozpoczęli partię - **zaczynał Żwirek**. Gra trwała długo, bo Żwirek i Muchomorek, jak na prawdziwych leśnych strategów przystało, liczyli, układali i analizowali. Legenda głosi, że zakończenie odcinka *Jak Żwirek i Muchomorek zjedli Lentilky* nigdy nie zostało wyemitowane, bo ani operator, ani reżyser nie mogli doczekać się rozwiązania długiej rozgrywki tej dziwnej gry. Z czasem taśma się zgubiła, a widzowie zapomnieli o całej sprawie.

Na szczęście możemy odkryć prawdę sami. **Znając układ cukiereczków na stosie oraz liczbę k (czyli ile cukierków można zdjąć w jednym ruchu), chcemy ustalić, czy zaczynający grę Żwirek wygrał**.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby N oraz k - odpowiednio wysokość stosu oraz liczba cukierków, które można zdjąć ze stosu w jednej rundzie.

W drugim wierszu znajduje się opis stosu. Jest to ciąg $A_1, \ldots, A_N$ liter N oraz C. Na szczycie stosu znajduje się element o największym indeksie (czyli skrajnie prawy kamień, na początku gry jest to A_N). Jeśli $A_i = N$, wówczas na i-tej pozycji od dołu stosu znajduje się niebieski cukierek; jeśli $A_i = C$, wówczas na tej pozycji znajduje się cukierek w kolorze czerwonym.

Wyjście

W jedynym wierszu wyjscia powinna znaleźć się odpowiedź na pytanie - *"Czy Żwirek może wygrać tę rundę, jeśli obydwaj gracze grają zgodnie najlepszą możliwą strategią a Żwirek wykonuje pierwszy ruch?"*.

Jeśli odpowiedź jest pozytywna, wypisz słowo TAK, w przeciwnym przypadku - NIE.

Ograniczenia

$1 \leq N \leq 10^6$,

$1 \leq k \leq N$.

Na stosie jest przynajmniej jeden czerwony cukierek.

Przykład

Wejście

```
5 1
CNNCN
```

Wyjście

```
TAK
```

Wyjaśnienie

W pierwszym ruchu Żwirek zbiera niebieskiego cukierka z góry stosu. Następuje ruch Muchomorka, który musi zdjąc cukierek w kolorze czerwonym. Żwirek i Muchomorek zbierają po jednym niebieskim cukierku. W ostatnim ruchu Żwirek zbiera ostatni czerwony cukierek, wygrywając.

Wejście

```
4 2
NCCN
```

Wyjście

```
NIE
```

Wyjaśnienie

Żwirek nie może wygrać tej gry.
Jeśli w pierwszym ruchu zdejmie jednego niebieskiego cukierka ze stosu, wówczas w kolejnym ruchu Muchomorek wygra, zdejmując dwa cukierki, w tym ostatniego czerwonego. Żwirek zje ostatniego niebieskiego cukierka.
NCCN –> N**CC** –> N –> pusty stos.
Jeśli w pierwszym ruchu Żwirek zdejmie dwa cukierki ze stosu, to w kolejnym ruchu Muchomorek wygra, zdejmując jednego lub dwa cukierki.
NCCN –> N**C** –> N –> pusty stos.
NCCN –> **NC** –> pusty stos.

Pomysłowy Dobromir (D)

Limit pamięci: 256 MB   Limit czasu: 1.00 s

Pewnego dnia Pomysłowy Dobromir postanowił skonstruować nowy wynalazek – maszynę do robienia tęczy! Wszystko było już prawie gotowe: zębatki się kręciły, woda pryskała, a promienie słońca odbijały się od lusterek. Niestety, aby maszyna mogła zadziałać, Dobromirowi brakowało jednego składnika - musiał użyć tajemniczego eliksiru tęczy, którego recepturę znalazł w starym zeszycie od chemii.

– Ach, te reakcje chemiczne! – westchnął Dobromir, wpatrując się w równania zapisane w zeszycie. – Muszę sprawdzić, czy te równania są zrównoważone, bo inaczej zamiast tęczy będzie tu zwykła mgiełka!

Z zapałem zaczął liczyć atomy w każdej formule – lewa strona, strzałka, prawa strona i znowu to samo. Dobromir pochylił się nad zeszytem, podrapał się po rudej czuprynie. Żaróweczka pojawiła się nad głową, jednak nie rozbłysła światłem. Dobromir nie był najlepszy w rachunkach – w końcu większość czasu spędzał na majsterkowaniu, a nie na nauce chemii.

I nagle - spojrzenie Dobromira z zeszytu przeniosło się w stronę kamery – może ktoś, kto ma więcej doświadczenia w liczeniu, pomoże mu rozwiązać tę zagadkę? Żaróweczka rozświetliła się. Tak oto Dobromir postanowił zwrócić się o pomoc do Ciebie! Rozwiąż problem Dobromira sprawdzając, czy równania chemiczne są zrównoważone, aby maszyna mogła w końcu stworzyć tęczę!

Zadanie Dobromira składa się z N równań reakcji chemicznych. Dla każdego z nich należy sprawdzić, czy jest ono *zrównoważone*. Równanie chemiczne jest symbolicznym zapisem przebiegu reakcji. W reakcji chemicznej pewien zestaw początkowych cząsteczek (*substraty*) reaguje, tworząc nowy zestaw cząsteczek (*produkty*).

Każde równanie składa się z lewej i prawej strony. Lewa strona zawiera wzory chemiczne substratów, natomiast prawa zawiera wzory produktów. Lewa i prawa strona równania są oddzielone strzałką (znakiem ->). Różne cząsteczki pojawiające się po lewej lub prawej stronie są rozdzielone znakiem +.

Cząsteczki zbudowane są z **atomów (połączonych wiązaniem) oznaczonych wielkimi literami alfabetu łacińskiego** (na potrzeby tego zadania). Wzór cząsteczki określają wszystkie występujące w nim atomy. Jeśli cząsteczka ma wiele wystąpień danego atomu, to ich liczba zapisywana jest **po symbolu atomu**. Na przykład $AC4B$ to wzór cząsteczki, która ma jeden atom A, 4 atomy C i jeden atom B.

Jeśli po jednej stronie równania cząsteczka pojawia się więcej niż raz, liczba tych cząsteczek jest zapisana jako **współczynnik przed wzorem**. Na przykład $3AC4B$ oznacza 3 cząsteczki $AC4B$, co daje w sumie 3 atomy A, 12 atomów C i 3 atomy B.

Równanie chemiczne uważa się za *zrównoważone*, jeśli prawa i lewa strona zawierają **równą liczbę atomów każdego rodzaju**. Twoim zadaniem jest określić, czy każde z N **równań chemicznych jest zrównoważone**.

Wszystkie współczynniki przed cząsteczkami oraz za atomami są pojedynczymi cyframi.

Wejście

W pierwszym wierszu wejścia znajduje się liczba N - liczba reakcji chemicznych. W kolejnych N wierszach wejścia znajdują się ciągi liter (każdy o długości ≤ 1000 znaków), określające równania reakcji chemicznych - po lewej stronie strzałki znajdują się substraty, po prawej produkty.

Wyjście

W N wierszach wyjścia należy wypisać TAK - jeśli równianie jest zrównoważone, albo NIE - jeśli nie jest.

Ograniczenia

$1 \leq N \leq 100$.

Każdy wiersz ma długość ≤ 1000 znaków.

Część punktów otrzymasz za rozwiązanie zadania dla równań, w których:

1. Nie ma współczynników i wszystkie cząsteczki są atomami,
2. Wszystkie cząsteczki są atomami.

Przykład

Wejście

```
2
H2+O2->H2O
2H2+O2->2H2O
```

Wyjście

```
NIE
TAK
```

Wyjaśnienie

W pierwszej reakcji liczba atomów O (tlenu) po lewej stronie to 2, natomiast po prawej stronie jest tylko 1 atom tlenu.
Druga reakcja jest zrównoważona - są 4 atomy H oraz 2 atomy O po obu stronach strzałki.

Wejście

```
5
2CH+3O2->2CO+H2O
2AO3C2+4AD->2CO2+4O2A+2D
BO2+H2SO4->BSO4+H2O2
H2O2->H2O+O
2CA+O2->2CO
```

Wyjście

```
NIE
NIE
TAK
TAK
NIE
```

Wyjaśnienie

W pierwszej reakcji liczba atomów O po lewej stronie to 6, a po prawej 3.
W drugiej reakcji nie zgadza się liczba atomów A, O, C oraz D.
Trzecia reakcja jest zrównoważona.
Czwarta reakcja również jest zrównoważona.
Piąta reakcja nie jest zrównoważona ze względu na brak atomu A po prawej stronie strzałki.

Mogiłkowo **(E)**

Limit pamięci: 256 MB   Limit czasu: 1.00 s

Mogiłkowo, 1 mila

– Miasteczko! Pójdziemy tędy! – z radością wykrzyknął Wirt, patrząc na litery wyryte w drewnianej tabliczce.

– Tak, kierunek Mogiłkowo – przytaknął Greg, trzymając pod pachą żabusia o imieniu Wirt Junior.

Bracia ruszyli w kierunku wskazywanym przez tabliczkę i wkrótce dotarli do cichego miasteczka... może nawet *zbyt* cichego.

– Słyszysz ten śpiew? – zapytał Greg, wskazując palcem na starą stodołę, z której dochodziły radosne głosy.

Kiedy chłopcy uchylili drewniane drzwi, ich oczom ukazały się postacie w dyniowych strojach, tańczące wokół wystrojonego słupa. W momencie, gdy weszli do środka, śmiechy i muzyka nagle ucichły, a wszyscy zwrócili się w stronę intruzów.

– Powiedzcie mi, chłopcy, jakim cudem trafiliście do Mogiłkowa? – zapytał największy z *dyniaków*, wystrojony w kapelusz z jesiennych liści.

– No... więc szukaliśmy drogi do domu. Przyszliśmy przez las i zobaczyliśmy wasze pola, no i pomyśleliśmy: *Hej, to zwyczajne miasteczko ze zwykłymi ludźmi*. A potem usłyszeliśmy śpiew w stodole... i... czy możemy już sobie iść? – Wirt mówił szybko, z nerwowym uśmiechem.

Największy dyniak westchnął.

– Bardzo mi przykro, dzieciaczki, ale za swoje występki będziecie musieli odpokutować. Z mocy prawa Mogiłkowej Izby Handlowej, uznaję was winnymi wtargnięcia, zakłócania porządku i... *morderstwa*.

– M-morderstwa?! – wykrzyknął przerażony Wirt.

– No dobra, przesadziłem z tym morderstwem – odparł dyniak, machając ręką. – Ale za pozostałe występki skazuję was na kilka godzin prac społecznych.

I tak, zgodnie z rozkazem mieszkańców, bracia zabrali się do pracy. W Mogiłkowie nic nie jest zwyczajne, więc ich zadanie również okazało się osobliwe. Greg i Wirt musieli **uporządkować stos trumienek**, w których dyniowi mieszkańcy odpoczywają po długich tańcach w stodole. W końcu każdy ma prawo do wygodnego relaksu, a dyniowe buty bywają ciężkie dla nóg...

Dane jest N trumienek, każda o wadze $w_1, w_2, ..., w_N$, ułożonych w stosie.

Bracia otrzymali listę M zamówień. Każde zamówienie wskazuje trumienkę a_i**, którą należy wyjąć ze stosu i przesunąć na wierzch**.

Aby zrealizować zamówienie na trumnę a_i, bracia muszą:

1. Podnieść wszystkie trumienki leżące **powyżej** trumny a_i, dźwigając ich **łączną wagę**.
2. Wyjąć trumnę a_i i przenieść ją na **szczyt stosu**.
3. Pozostałe podniesione trumny odłożyć **w tej samej kolejności**, co wcześniej.
4. Każda trumna może być zamawiana dowolnie wiele razy.
5. Początkowe ułożenie stosu może być **dowolne**.

Pomóż Wirtowi i Gregowi tak ułożyć początkowy stos, aby **suma mas** podniesionych trumien podczas realizacji wszystkich zamówień była **jak najmniejsza**.

Wejście

W pierwszym wierszu wejścia znajdują się dwie liczby, N ($2 \leq N \leq 500$) oraz M ($1 \leq M \leq 1000$), oddzielone spacją.

W drugim wierszu znajduje się ciąg wag: $w_1, \ldots, w_N$ ($1 \leq w_i \leq 100$).

W trzecim, ostatnim wierszu wejścia znajduje się ciąg zamówień: $a_1, \ldots, a_M$ ($1 \leq a_i \leq N$).

Wyjście

W jedynym wierszu wyjścia powinna się znaleźć **minimalna suma wag** podniesionych przez chłopców trumienek.

Ograniczenia

$2 \leq N \leq 500$,
$1 \leq M \leq 1000$,
$1 \leq w_i \leq 100$.

Przykład

<table>
<tr><th>Wejście</th><th>Wyjście</th><th>Wyjaśnienie</th></tr>
<tr><td><pre>4 5
10 30 20 40
2 1 1 3 2</pre></td><td><pre>100</pre></td><td>Początkowa konfiguracja stosu: $2\ 1\ 3\ 4$ (lewa strona to szczyt stosu).
Najpierw zdejmą trumnę numer 2 ze szczytu stosu, kosztem 0 i od razu odłożą ją z powrotem na szczyt.
Następnie przeniosą trumnę 1 na szczyt, podnosząc trumnę 2 o wadze 30.
Kolejne podniesienie trumny 1 nic nie kosztuje.
Następnie, aby przenieść trumnę 3 na wierzch, podniosą trumny 1 i 2, które ważą łącznie 40.
Poniesienie trumny 2 wymaga podniesienia trumien 1 i 3, o łącznej wadze 30.
Wybrana przez Wirta i Grega początkowa konfiguracja jest optymalna, co można sprawdzić.</td></tr>
</table>

<table>
<tr><th>Wejście</th><th>Wyjście</th><th>Wyjaśnienie</th></tr>
<tr><td><pre>5 2
10 3 7 12 5
3 3</pre></td><td><pre>0</pre></td><td>Wirt i Greg mogą ułożyć stos w taki sposób: $3\ 2\ 4\ 5\ 1$, dzięki czemu nie będą musieli dźwigać pozostałych trumien.</td></tr>
</table>

Zbanowany bez powodu **(F)**

Limit pamięci: 512 MB Limit czasu: 2.00 s

Johnny, niepoprawny romantyk, znowu narobił bałaganu w swoim życiu uczuciowym. Uwielbia flirtować i sypać tandetnymi komplementami, co – o dziwo – czasami działa. Niestety, Johnny ma pamięć krótszą niż... zarost na jego gładkim licu. Jedyną rzeczą, jaką zapisuje w swoim telefonie, są numery telefonów dziewczyn oraz pierwsze litery ich imion.

Ale jest problem. Niektóre dziewczyny zablokowały Johnny'ego (i nie bez powodu). Kiedy tak się dzieje, numer w jego telefonie pozostaje bez litery, a Johnny zupełnie go ignoruje.

Dziś są Walentynki – najważniejszy dzień w życiu Johnny'ego. Marzy o tym, by umówić się na randki z kilkoma dziewczynami. Jednak jego sposób wyboru jest, delikatnie mówiąc, niecodzienny:

1. Johnny wybiera spójny przedział ze swojej listy N numerów, np. przedział od 3 do 7.
2. Następnie patrzy na litery przypisane do numerów w tym przedziale (włącznie z końcami przedziału). Jeśli jakaś litera pojawia się **więcej niż raz**, Johnny jest zbyt zagubiony, by zadzwonić (nie chce popełnić faux pas, myląc imiona!). Ale jeśli **wszystkie litery w przedziale są unikatowe**, wówczas Johnny może swobodnie zadzwonić do którejś z Pań.
3. Niestety, wraz z upływem dnia Johnny jest blokowany przez kolejne dziewczyny. Kolejność *banów* jest dokładnie określona: i-ta dziewczyna, która go blokuje, ma numer p_i na liście Johnny'ego.

Twoim zadaniem jest pomóc Johnny'emu i odpowiedzieć na pytanie: po ilu banach Johnny będzie mógł bez wątpliwości zadzwonić do dziewczyn w Q określonych przedziałach?

Wejście

W pierwszym wierszu wejścia znajduje się N literowe słowo - pierwsze litery imion dziewczyn, których numery zapisane ma Johnny.

W drugim wierszu znajduje się liczba Q - liczba przedziałów.

W i-tym z kolejnych Q wierszy znajdują się liczby a_i, b_i, oznaczające lewy i prawy koniec i-tego przedziału ($1 \leq a_i \leq b_i \leq N$).

W ostatnim wierszu znajduje się **permutacja** N elementów - ciąg p_i ($1 \leq i \leq N$), określający kolejność blokowania Johnnego przez dziewczyny (permutacja to taki ciąg liczb od 1 do N, w którym każda liczba występuje dokładnie raz).

Wyjście

W jedynym wierszu wyjścia powinna znaleźć się **minimalna** liczba dziewczyn, które muszą zablokowac Johnny'ego, aby ten mógł bezpiecznie umówić się na randki.

Ograniczenia

$1 \leq N \leq 10^5$,

$1 \leq Q \leq 10^5$.

W tym zadaniu można otrzymać część punktów za wolniejsze rozwiązania (dla mniejszych N i Q).

Przykład

Wejście

```
abcbd
2
1 4
4 5
3 2 5 1 4
```

Wyjście

```
2
```

Wyjaśnienie

Po drugim blokowaniu lista kontaktów Johnny'ego wygląda następująco $a**bd$. Po pierwszym blokowaniu lista wyglądała tak: $ab*bd$, a Johnny nie mógł zdecydować się na wybór dziewczyny w przedziale od 1 do 4 (dwukrotnie widzi

Kombinatoryczny Żabuś (G)

Limit pamięci: 256 MB Limit czasu: 1.00 s

-Ten świat jest padołem łez, Greg. Życie to nie zabawa. - Beatrice, niebieski ptak, spojrzała z góry na Grega, który niósł swoją żabkę pod pachą. Jej głos był pełen pouczenia i troski. Po kilku minutach jej monologu... Greg już zniknął. - Hej, gdzie jest Greg?

-Och. Uh, pewnie gdzieś sobie poszedł – odparł spokojnie Wirt, poprawiając kapelusz.

Tymczasem Greg biegł przez las z dzbankiem i żabką na głowie, z energią, która mogłaby zawstydzić wiatr.

-Musimy robić swoje, aby ten świat był lepszy! - krzyczał, przeskakując przez korzenie.

-Raur – odpowiedziała mu żabka.

Greg zatrzymał się, gdy usłyszał cichą melodię pianina, dobiegającą z budynku pośród drzew.

-Jeju! Szkoła?! Tylko nie dzisiaj! – wrzasnął w panice i uciekł w stronę pobliskiej rzeczki, gdzie zauważył grupę zwierzątek – wagarowiczów, nudzących się w cieniu drzew.

Po krótkiej rozmowie z nowymi towarzyszami, Greg szybko zorientował się, że rozmowa z nimi ma mniej treści, niż rozmowa z dzbanuszkiem na głowie. Żaden z nich nie słyszał o palindromach, nie mówiąc już o kombinatoryce. Greg, nieco zmartwiony, uznał, że świat nie może być tak mdły i nijaki, i postanowił nauczyć swoich nowych znajomych czegoś nowego. Zaczął mówić o swojej ulubionej grze *w dwa stare kocury*. Ale zaraz przerwał. *To nie ta gra.* Wrócił więc do sedna i wyjaśnił, że ich zadanie jest proste: wymyślić imię dla jego żabki. Żadnych prostych słów, żadnych banalnych propozycji.

Ale Greg po zrobieniu zadania A, B, i wszystkich pozostałych, nie chciał już stawiać na prostotę. Palindromy w ich klasycznej formie? Nuuuda. Tym razem wpadł na pomysł, by wypróbować formy angażującej wszystkich jego M kolegów do zabawy. Imie dla żabusia powinno być N-literowym słowem, które korzysta z alfabetu o K różnych literach. Każdy z M wagarowiczów wskazuje parę indeksów (l_i, r_i), określając, że podciąg imienia od pozycji l_i **do** r_i włącznie musi być **palindromem**.

Dla przypomnienia, *palindromem* nazywamy słowo, które czytane od lewej do prawej i od prawej do lewej, jest takie samo.

Ale zaraz... - pomyslał Greg. - Ile w ogóle jest takich imion?

Twoim zadaniem jest policzenie, **ile różnych imion, spełniających te warunki, Greg może wymyślić dla swojej żabki**.

Wejście

W pierwszym wierszu wejścia znajdują się trzy liczby - N, K oraz M ($1 \leq N, K \leq 2000$, $0 \leq M \leq 2000$).

W kolejnych M wierszach wejścia znajdują się opisy dane przez kolegów: w i-tym ($1 \leq i \leq M$) wierszu znajduje się para l_i, r_i ($1 \leq l_i \leq r_i \leq N$).

Wyjście

W jedynym wierszu wyjścia powinna się znaleźć liczba sposobów, na które można stworzyć słowo spełniające warunki Grega, modulo $10^9 + 7$.

Ograniczenia

$1 \leq N, K \leq 2000$,

$0 \leq M \leq 2000$.

Niektóre testy spełniają warunek $M = 0$.

Przykład

Wejście	Wyjście	Wyjaśnienie
5 3 2 1 3 2 5	9	Greg może ułożyć nastepujące imiona (korzystając dla przykładu z trzech pierwszy liter z alfabetu): aaaaa, abaab, acaac, babba, bbbbb, bcbbc, cacca, cbccb, ccccc.