Zadanie A: Aliasy

Limit czasowy: 40s, limit pamięciowy: 1GB.

Do Akademickich Mistrzostw Polski w Programowaniu Indywidualnym na Czas (AMPPIC) zgłosiło się n zawodników. Twoim zadaniem jest założyć wszystkim konta. Każde konto będzie miało unikalny *alias*, czyli nazwę użytkownika. System AMPPIC przyjmuje tylko aliasy złożone z małych liter alfabetu angielskiego i cyfr.

Aby zachować porządek, masz zwyczaj tworzyć aliasy na podstawie imion i nazwisk zawodników, zawsze według tego samego przepisu: a pierwszych liter imienia, potem b pierwszych liter nazwiska, a na końcu c wybranych przez Ciebie cyfr. Jeżeli imię liczy sobie mniej niż a liter lub nazwisko mniej niż b liter, bierzesz po prostu wszystkie litery (wtedy alias jest krótszy, ale to nie problem). Na przykład dla $a = b = c = 3$ James Bond mógłby dostać alias `jambon007`, a Lady Di – `laddi123`.

Chcesz, aby aliasy były jak najkrótsze – to zaoszczędzi kilka bajtów pamięci, a Ty lubisz oszczędzać pamięć. Z drugiej strony, wszyscy zawodnicy muszą mieć unikalne loginy. Dla danej listy użytkowników (ich imion i nazwisk) znajdź takie a, b i c, które pozwolą na stworzenie różnych loginów dla wszystkich zawodników, i dla których $a + b + c$ będzie możliwie najmniejsze. Liczby a, b i c mogą być równe 0 (ale nie wszystkie jednocześnie – loginy muszą być niepuste).

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 6$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera jedną liczbę całkowitą n ($1 \leqslant n \leqslant 200\,000$) – liczbę zawodników. W kolejnych n liniach znajdują się imiona i nazwiska zawodników – zarówno imię, jak i nazwisko jest ciągiem złożonym wyłącznie z małych liter alfabetu angielskiego.

Suma długości wszystkich imion i wszystkich nazwisk zawodników w jednym zestawie testowym nie przekracza $1\,500\,000$. Jak najbardziej możliwe jest, że dwoje zawodników nazywa się tak samo (i wciąż muszą oni otrzymać różne aliasy).

Wyjście

Dla każdego zestawu danych wypisz jedną linię zawierającą trzy liczby całkowite nieujemne a, b i c, które pozwalają na stworzenie różnych aliasów i mają najmniejszą możliwą sumę. Jeśli jest kilka możliwych rozwiązań, wypisz dowolne z nich.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
1 11 sven eriksson erik svensson sven svensson erik eriksson bjorn eriksson bjorn svensson bjorn bjornsson erik bjornsson sven bjornsson thor odinsson odin thorsson	1 1 0

Wyjaśnienie

Różne aliasy można otrzymać biorąc inicjały wszystkich zawodników (pierwsze litery imienia i nazwiska – `se`, `es`, `ss`, `ee`, ...). Odpowiedź `1 0 1` również byłaby poprawna, jako że da się stworzyć różne aliasy dopisując jedną cyfrę do pierwszej litery imienia każdego zawodnika.

Zadanie B: Bary

Limit czasowy: 6s, limit pamięciowy: 1GB.

Bycie burmistrzem wsi Prostów Dolny to nie lada wyzwanie. Wprawdzie wydatki na rozwój infrastruktury drogowej są minimalne – domy wszystkich n mieszkańców, ponumerowane liczbami od 1 do n, leżą w kolejności wzdłuż jednej prostej drogi ciągnącej się przez całą wieś – ale jednak czasem trzeba podejmować trudne decyzje. Na przykład takie, jak wydawanie pozwoleń na otwarcie baru.

Okazuje się, że wszyscy mieszkańcy Prostowa marzą o otwarciu własnego baru. Na Twoje biurko trafiło właśnie n formularzy o pozwolenie, po jednym na mieszkańca. Każdy z mieszkańców przedstawił swój biznesplan, z którego najbardziej interesuje Cię proponowana wysokość odprowadzanego podatku. Jeśli bar zostanie otwarty, i-ty mieszkaniec obiecuje płacić wsi p_i złotych monet od każdego klienta.

Planujesz udzielić pozwolenia na otwarcie barów pewnemu (niepustemu) podzbiorowi mieszkańców (być może nawet wszystkim). Każdy mieszkaniec, niezależnie od otrzymania pozwolenia na otwarcie własnego baru, zostanie klientem dwóch innych barów: najbliższego ściśle po lewej od swojego domu i najbliższego ściśle po prawej (chyba że któryś z nich nie istnieje – wtedy będzie klientem odpowiednio mniejszej liczby placówek). Wyznaczając najbliższe bary, nikt nie bierze pod uwagę swojego własnego, jako że nawet najlepszy barman nie powinien przecież obsługiwać siebie samego. Po rozstrzygnięciu które z barów zostaną otwarte, każdy zacznie generować wpływ do budżetu wsi wynoszący p_i za każdego klienta. Przykładowo, jeśli $n = 5$ i otwarte zostaną trzeci oraz piąty bar, to pierwszy będzie mieć 4 klientów, a drugi 2 klientów, generując wpływ z podatków wynoszący $4p_3 + 2p_5$.

Znając obiecaną wysokość podatku odprowadzanego od każdego z hipotetycznych barów, wyznacz maksymalny zysk, jaki możesz osiągnąć wydając pozwolenia w optymalny sposób.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 10\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera jedną liczbę całkowitą n ($2 \leqslant n \leqslant 500\,000$) – liczbę mieszkańców Prostowa Dolnego.

Druga linia zestawu zawiera n liczb całkowitych p_i ($1 \leqslant p_i \leqslant 10^9$) – proponowaną wysokość podatku od klienta dla każdego z n hipotetycznych barów.

Sumaryczna liczba mieszkańców wsi we wszystkich przypadkach testowych nie przekroczy $3 \cdot 10^6$.

Wyjście

Dla każdego zestawu danych wypisz, w osobnej linii, jedną liczbę całkowitą, oznaczającą maksymalny sumaryczny zysk jaki możesz osiągnąć, odpowiednio wydając pozwolenia na otwarcie barów.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
2 4 5 2 2 6 5 1 5 4 4 1	33 29

Wyjaśnienie

W pierwszym teście przykładowym optymalnym rozwiązaniem jest pozwolić na otwarcie pierwszego i ostatniego baru. Każdy z nich będzie mieć 3 klientów, co daje $3 \cdot 5 + 3 \cdot 6 = 33$ monety zysku.

W drugim teście przykładowym optymalnie jest pozwolić na wszystkie bary poza trzecim, co generuje zysk $1 \cdot 1 + 3 \cdot 5 + 3 \cdot 4 + 1 \cdot 1 = 29$. Gdyby zamiast tego pozwolić na otwarcie wszystkich barów zysk byłby mniejszy – 28 monet.

Zadanie C: Ctrl+C Ctrl+V

Limit czasowy: 5s, limit pamięciowy: 1GB.

– *Jak tam u ciebie z tą autobiografią?*
– *Ee, Aniu, co? Było jakieś zadanie na polski?*
– *No tak, napisać autobiografię. Zapomniałaś?*
– *Zapomniałam. Dasz odpisać?*
– *Autobiografię chcesz odpisać!? No dobra, ale weź trochę pozmieniaj.*

Masz dane słowo s złożone z małych liter angielskiego alfabetu. Słowo to jest autobiografią napisaną przez Anię, co oznacza, że może zawierać słowo `ania` jako spójne podsłowo, być może wielokrotnie. Wyznacz minimalną liczbę znaków, jaką należy zmienić w s, tak aby nie zawierało spójnego podsłowa `ania`.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 10\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza i jedyna linia zestawu zawiera słowo s – autobiografię Ani. Słowo ma długość l ($1 \leqslant l \leqslant 10^6$) i składa się z małych liter angielskiego alfabetu.

Całkowita długość słów we wszystkich zestawach nie przekracza $5\,000\,000$.

Wyjście

Dla każdego zestawu danych, w osobnej linii, wypisz jedną liczbę całkowitą, oznaczającą minimalną liczbę zmian, jakich trzeba dokonać w słowie, aby nie zawierało spójnego podsłowa `ania`.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
3 aniasieurodzilaapotemnicsieniedzialo nicciekawegouanianiagnieszkianialicji jeszczekrotszaautobiografiaani	1 2 0

Zadanie D: Droga na szczyt

Limit czasowy: 10s, limit pamięciowy: 1GB.

Najwyższym szczytem pasma Ramtopów jest Mount Stackframe – niezwykle malowniczo położona góra słynąca z pięknych widoków. Na Mount Stackframe wybudowano n schronisk (numerowanych od 1 do n), a także $n-1$ odcinków szlaku, z których każdy łączy dwa schroniska. Schronisko numer 1 położone jest na samym szczycie góry i wiadomo, że z każdego innego schroniska da się dojść do 1 na dokładnie jeden sposób (o ile po drodze nie zawracamy). Niektóre schroniska są położone u podnóża góry – to te, z których wychodzi dokładnie jeden szlak (nie dotyczy to schroniska 1). Tradycyjna wycieczka na Mount Stackframe zaczyna się w schronisku u podnóża i kończy na szczycie w schronisku 1. Jeśli turysta zatrzyma się w pewnym schronisku x i popatrzy w dół, to zobaczy pewną liczbę schronisk – są to dokładnie te, z których droga do 1 prowadzi przez x. Nowy projekt organizacji parku narodowego zakłada, że jeśli liczba widocznych z x schronisk (wliczając x) wynosi dokładnie d, to w schronisku zostanie wybudowany punkt widokowy.

Znajdź wszystkie wartości d, dla których każdy turysta wędrujący od podnóża góry na jej szczyt, odwiedzi co najmniej jeden punkt widokowy.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 10\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera jedną liczbę całkowitą n ($2 \leqslant n \leqslant 10^6$) – liczbę schronisk. Każda z kolejnych $n-1$ linii zawiera dwie liczby całkowite a_i i b_i ($1 \leqslant a_i, b_i \leqslant n$), oznaczające że i-ty szlak łączy schroniska a_i i b_i. Szlaki nie krzyżują się poza schroniskami.

Sumaryczna liczba schronisk we wszystkich zestawach testowych nie przekroczy $3 \cdot 10^6$.

Wyjście

Dla każdego zestawu danych wypisz dwie linie: pierwsza powinna zawierać liczbę możliwych wartości d, dla których warunek z zadania jest spełniony (tj. każdy turysta przejdzie przez jakiś punkt widokowy). Druga linia powinna zawierać te wartości podane w kolejności rosnącej.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
1 9 1 2 2 3 3 4 3 5 2 6 6 7 7 8 7 9	4 1 3 8 9

Zadanie E: Euklides

Limit czasowy: 30s, limit pamięciowy: 8MB.

Siedzisz na egzaminie i zastanawiasz się, jak wyznaczyć największy wspólny dzielnik. Najwyższym wysiłkiem przypominasz sobie, że $\texttt{gcd}(x, y)$ to największa liczba całkowita $d \geqslant 1$, która dzieli zarówno x jak i y. Mimo dziur w pamięci kojarzysz, że coś takiego pojawiło się na wykładzie. Niestety, nawet jeśli Twój umysł zarejestrował podczas zajęć jakieś strzępki informacji, to zostały one dawno wyparte przez przygody Wiedźmaka Gebajta i najnowsze odcinki Peaky Bajters...

Zaczynało się od x oraz y i dopisywało na tablicy jakieś różnice wcześniej zapisanych liczb? Odganiasz nieprzyjemną myśl sugerującą, że znowu będzie tak samo jak rok temu z medianą... A może tamto to był tylko zły sen?

Na bazie mglistych wspomnień z wykładu wymyśliłeś następujący algorytm. Na początku na tablicy napisane są liczby x oraz y. W jednym ruchu można wybrać dwie liczby a i b, które znajdują się na tablicy, i dopisać $c = 2a - b$, o ile $c > 0$. Wynik dla pary (x, y) to najmniejsza liczba, jaką da się uzyskać powtarzając powyższą operację pewną liczbę razy (być może zero).

Po sprawdzeniu na kilku testach algorytm wyglądał obiecująco. Dla pary $(10, 14)$, w pierwszym ruchu możemy zapisać na przykład liczbę 6 (ponieważ $2 \cdot 10 - 14 = 6$), zaś w kolejnym 2 (ponieważ $2 \cdot 6 - 10 = 2$). Z drugiej strony, można pokazać że nie da się zapisać liczby 1, a więc algorytm zwrócił poprawną odpowiedź. Niestety, dla $(10, 16)$ najmniejszą liczbą jaką udało Ci się uzyskać jest 4, zaś poprawną odpowiedzią jest 2. Coś tu nie gra...

Dla danego n wyznacz liczbę par (x, y) $(1 \leqslant x < y \leqslant n)$ dla których Twój algorytm zwraca poprawną wartość $\texttt{gcd}(x, y)$ [1].

Zwróć uwagę, że to zadanie ma niski limit pamięci – 8MB.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z $(z \geqslant 1)$. Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza i jedyna linia zestawu zawiera jedną liczbę całkowitą n $(n \geqslant 2)$ o znaczeniu wyjaśnionym w treści zadania.

Testy

Każdy plik testowy należy do jednej z trzech poniższych grup:

- $z \leqslant 3000$, $n \leqslant 10^6$
- $z = 30$, $n \leqslant 10^9$
- $z = 3$, $n \leqslant 10^{11}$

Wyjście

Dla każdego zestawu danych wypisz, w osobnej linii, jedną liczbę całkowitą, oznaczającą szukaną liczbę par.

[1] W szczególności, jeśli $\texttt{gcd}(x, y) = x$, to uznajemy że algorytm zadziałał poprawnie, bo x znajduje się już na tablicy.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
3 2 5 14	1 9 62

Zadanie F: Fiołki i róże

Limit czasowy: 30s, limit pamięciowy: 1GB.

Przed pałacem Bytingham znajduje się piękny ogród. Co roku sprowadza on przed królewski dwór rzesze podróżnych, chcących zobaczyć na własne oczy jeden z cudów świata. Król Intles III przez lata inwestował przede wszystkim w długość ogrodu, dzięki czemu w jednym rzędzie można posadzić aż $3n$ kwiatów.

Obecny ogrodnik, który wiele energii poświęcił temu majestatycznemu zielonemu przedmiotowi królewskiej dumy, postanowił ostatnio przejść na wczesną emeryturę jeszcze przed czterdziestką. Właśnie przyjechałeś do pałacu, aby przejąć jego rolę – i choć widok twarzy Twojego poprzednika sprawił, że zacząłeś kwestionować swoje zdolności określania wieku innych ludzi, to ochoczo zaakceptowałeś ofertę. Teraz czeka Cię pierwsze zadanie!

W tym roku król Intles postanowił, że w ogrodzie posadzone zostaną dwa typy kwiatów: fiołki i róże. Muszą one jednak spełniać odpowiedni schemat, określony wielostronicowym królewskim dekretem. Na pierwszej stronie znajduje się napis:

Miejsca na fiołki oraz róże zostały ponumerowane liczbami od 1 *do* $3n$.

Wszystkie następne są bardzo podobnej postaci:

Co najmniej jeden spośród następujących warunków musi zostać spełniony:
- *Wszystkie rośliny posadzone na miejscach od* a_i *do* b_i *włącznie mają być różami.*
- *Wszystkie rośliny posadzone na miejscach od* c_i *do* d_i *włącznie mają być fiołkami.*

Zdumiony czytasz q stron z prawie identycznymi poleceniami, różniącymi się jedynie liczbami a_i, b_i, c_i, d_i. Na razie nie brzmi to źle, ale na ostatniej stronie zauważasz jeszcze jeden przerażający napis:

Dostępnych mamy dokładnie $2n$ *róż oraz* $2n$ *fiołków.*

W jednej chwili przypominasz sobie twarz ogrodnika, którego widziałeś przyjeżdżając do królewskiego pałacu, po czym łapiesz się za głowę. Czy to zadanie jest w ogóle wykonalne? Znajdź odpowiednie ustawienie kwiatów, lub ustal, że takie nie istnieje (i zacznij myśleć, jak uniknąć gniewu króla).

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z $(1 \leqslant z \leqslant 10^5)$. Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii znajdują się dwie liczby całkowite n oraz q $(1 \leqslant n \leqslant 33\,333, 1 \leqslant q \leqslant 10^5)$.

Kolejnych q linii zestawu zawiera opisy wytycznych z królewskiego dekretu. Linia numer i zawiera cztery liczby a_i, b_i, c_i, d_i $(1 \leqslant a_i \leqslant b_i \leqslant 3n, 1 \leqslant c_i \leqslant d_i \leqslant 3n)$ o znaczeniu opisanym w treści zadania.

Suma liczb n we wszystkich zestawach nie przekracza $333\,333$. Suma liczb q we wszystkich zestawach nie przekracza 10^6.

Wyjście

Dla każdego zestawu w pierwszej linii wypisz pojedynczy napis `TAK`, jeżeli możliwe jest zasadzenie wszystkich kwiatów zgodnie z wytycznymi, albo `NIE` w przeciwnym wypadku.

Jeżeli odpowiedź jest twierdząca, w drugiej linii wypisz napis o długości $3n$, złożony z liter `F` oraz `R`. Litera `F` na i-tej pozycji tego napisu oznacza, że i-tą rośliną posadzoną w rzędzie powinien być fiołek, zaś litera `R` oznacza różę. Napis ten nie może zawierać więcej niż $2n$ liter `F` ani więcej niż $2n$ liter `R`.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
2 1 3 1 1 2 2 1 2 3 3 1 1 3 3 1 3 1 1 2 2 2 2 3 3 3 3 1 1	TAK RFF NIE

Zadanie G: Gonitwa

Limit czasowy: 5s, limit pamięciowy: 1GB.

Zabawa w *policjantów i złodzieja* odbywa się na długiej ulicy, którą na potrzeby zadania możemy utożsamiać z osią liczbową. Jeden z uczestników (złodziej) ustawia się na pozycji 0 na osi, zaś pozostałych n osób (policjanci) po obu jego stronach (po każdej przynajmniej jeden). Wraz z rozpoczęciem zabawy każdy policjant zaczyna biec w stronę złodzieja z ustaloną prędkością, zaś złodziej rozpoczyna swoją ucieczkę z prędkością v, większą od prędkości każdego z policjantów, kierując się w prawo (zgodnie z rosnącymi wartościami na osi). Za każdym razem, gdy złodziej dobiegnie do pierwszego policjanta pędzącego na niego, zawraca (w pomijalnie małym czasie) i biegnie dalej w przeciwnym kierunku. Taka sytuacja powtarza się aż do spotkania pewnych dwóch policjantów, biegnących w przeciwnych kierunkach, kiedy to złodziej zostaje złapany i zabawa dobiega końca.

Dla danego układu początkowego, wyznacz, jaką odległość przebiegnie złodziej w czasie całej zabawy.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 10\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n ($2 \leqslant n \leqslant 400\,000$) i v ($1 < v \leqslant 10^6$) – liczbę policjantów i prędkość złodzieja.

Każda z kolejnych n linii zestawu zawiera dwie liczby całkowite p_i ($-10^{12} \leqslant p_i \leqslant 10^{12}$, $p_i \neq 0$) i v_i ($1 \leqslant v_i < v$) – początkową pozycję i prędkość i-tego policjanta.

Sumaryczna liczba policjantów we wszystkich zestawach testowych nie przekroczy $2 \cdot 10^6$.

Wyjście

Dla każdego zestawu danych, w osobnej linii, wypisz jedną liczbę rzeczywistą w formacie z kropką dziesiętną (nie w notacji naukowej), oznaczającą długość trasy pokonanej przez złodzieja. Aby odpowiedź została uznana za poprawną wystarczy, by błąd względny lub bezwzględny nie przekraczał 10^{-8}. Innymi słowy, jeśli Twój algorytm odpowie a, zaś poprawna odpowiedź to b, to wystarczy, by zachodziło $\frac{|a-b|}{\max(1,b)} \leqslant 10^{-8}$.

Wypisana liczba może mieć co najwyżej 20 cyfr po kropce dziesiętnej.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
3 4 9 10 2 -7 2 -6 1 7 1 2 8 -1 7 1 6 2 3 -1000000000000 1 1000000000000 1	38.25 1.23076923 3000000000000

Puste linie w wejściu testu przykładowego zostały dodane dla czytelności. Nie są one obecne w plikach testowych, na których uruchamiane będzie Twoje rozwiązanie.

Uwaga

Zauważ, że w tym zadaniu bierzemy pod uwagę zarówno błąd bezwzględny, jak i względny. W szczególności oznacza to, że gdy poprawna odpowiedź jest duża, dozwolony błąd Twojego programu również jest duży. W ostatnim teście przykładowym poprawną odpowiedzią jest $3 \cdot 10^{12}$, a więc dowolna liczba różniąca się od niej o nie więcej niż 30 000 byłaby zaakceptowana.

Zadanie H: Hyperloop

Limit czasowy: 50s, limit pamięciowy: 64MB.

Jest rok 2077. Sztuczna inteligencja rozwiązała właśnie ostatni problem kolorowania hipergrafów, a ekscentryczny miliarder Melon Usk zlecił świeżo pozbawionym zatrudnienia pracownikom Informatyki Analitycznej stworzenie oprogramowania dla kompleksu Hyperloop – systemu superszybkich połączeń międzymiastowych.

Z powodu rosnącego poziomu wód, jedyną masą lądową pozostałą na Ziemi jest pojedyncza wyspa, na której brzegu leży n ocalałych miast, nazwanych dla uproszczenia 1, 2, ..., n w kolejności występowania wzdłuż brzegu. Melon jest właścicielem promów kursujących wokół wyspy, a więc pomiędzy miastami i i $i+1$ dla $1 \leqslant i < n$ oraz pomiędzy miastami n i 1, i połączeń Hyperloop, łączących pewne pary *niesąsiednich* miast [1]. Pozostało jedynie zaprojektowanie oprogramowania wyznaczającego najkrótsze trasy pomiędzy miastami; w wersji beta będzie ono obsługiwać jedynie podróż z miasta 1 do miasta n.

Zadanie mogłoby zdać się trywialne: wszak oprogramowanie ma brać pod uwagę oba rodzaje połączeń, a między miastami 1 i n istnieje już *bezpośrednie* połączenie za pomocą promu, jednak wcale nie musi być ono najszybsze! Do tego może istnieć wiele najkrótszych ścieżek. W takiej sytuacji system powinien preferować ścieżki, w których skład wchodzą jak najdłuższe połączenia (im dłużej trwa pojedyncze połączenie, tym pasażer może być bardziej skłonny do zamówienia w jego trakcie kawy, pomnażając majątek Uska). Formalnie, aby porównać ścieżki **o tej samej sumarycznej długości**, należy wypisać długości występujących w nich połączeń (uwzględniając powtórzenia), posortować każdy z ciągów nierosnąco, i wybrać najpóźniejszy leksykograficznie [2].

Mając dany opis promów kursujących wokół wyspy oraz połączeń Hyperloop, znajdź optymalną ścieżkę z miasta 1 do miasta n. Podczas porównywania ścieżek oba rodzaje połączeń traktowane są w taki sam sposób, a wszystkie połączenia są dwukierunkowe.

Zwróć uwagę, że to zadanie ma niski limit pamięci – 64MB.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 600$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n oraz m ($3 \leqslant n \leqslant 100\,000$, $n \leqslant m \leqslant 300\,000$) – liczbę miast oraz liczbę połączeń.

Każda z kolejnych m linii zestawu zawiera trzy liczby całkowite u_i, v_i, d_i ($1 \leqslant u_i \neq v_i \leqslant n$, $1 \leqslant d_i \leqslant 50\,000$), oznaczające kolejno: numery miast które łączy dane dwukierunkowe połączenie, oraz czas potrzebny na jego przebycie.

Wśród połączeń podanych na wejściu pojawią się $(1,2)$, $(2,3)$, ..., $(n,1)$, odpowiadające połączeniom używającym promu. Każdą parę miast łączyć będzie co najwyżej jedno połączenie.

Sumaryczna liczba miast oraz liczba połączeń we wszystkich przypadkach testowych nie przekroczy odpowiednio $400\,000$ i $800\,000$.

[1] Połączenia Hyperloop używają podziemnych tuneli. Tunele, które mogłyby się przeciąć, zostały zwyczajnie poprowadzone na różnych głębokościach.

[2] Ciąg $a_1, ..., a_p$ jest wcześniejszy leksykograficznie od ciągu $b_1, ..., b_q$ jeśli jest jego prefiksem lub istnieje takie $i \leqslant \min(p, q)$ że $a_j = b_j$ dla $j < i$ oraz $a_i < b_i$. W tym zadaniu porównywane ciągi mają zawsze tę samą sumę, a więc podczas porównywania dwóch różnych ciągów żaden z nich nie będzie prefiksem drugiego.

Wyjście

Dla każdego zestawu danych, w pierwszej linii wypisz jedną liczbę całkowitą k, oznaczającą liczbę miast na optymalnej ścieżce. W drugiej linii wypisz k różnych liczb całkowitych, oznaczających numery kolejnych miast. Wyznaczona ścieżka powinna zaczynać się w mieście 1, kończyć w mieście n, i być optymalna według definicji z treści zadania. Jeśli istnieje wiele optymalnych ścieżek, możesz wypisać dowolną z nich.

Przykład

<table>
<tr><th>Dla danych wejściowych:</th><th>Możliwą poprawną odpowiedzią jest:</th></tr>
<tr><td><pre>
2
4 6
1 2 1
1 3 2
2 3 1
2 4 2
3 4 1
1 4 4
6 11
1 2 9
2 3 12
3 4 3
4 5 5
5 6 10
6 1 22
2 4 9
3 6 1
4 6 5
2 5 2
3 5 8
</pre></td><td><pre>
3
1 2 4
5
1 2 5 3 6
</pre></td></tr>
</table>

Wyjaśnienie

W pierwszym teście przykładowym minimalna odległość z miasta 1 do miasta 4 wynosi 3, i realizują ją trzy ścieżki: $1 \to 2 \to 4$, $1 \to 3 \to 4$ i $1 \to 2 \to 3 \to 4$. Według definicji z treści zadania pierwsze dwie ścieżki są lepsze od trzeciej, bo w porządku leksykograficznym $(2, 1) > (1, 1, 1)$. Dowolna z dwóch pierwszych ścieżek jest poprawną odpowiedzią. Ścieżka bezpośrednia $1 \to 4$ nie jest rozważana, gdyż ma długość 4.

W drugim teście przykładowym minimalną odległość realizują ścieżki $1 \to 2 \to 5 \to 3 \to 6$ i $1 \to 2 \to 5 \to 4 \to 3 \to 6$. Optymalna jest pierwsza z nich, gdyż porównując ich posortowane ciągi odległości dostajemy $(9, 8, 2, 1) > (9, 5, 3, 2, 1)$.

Zadanie I: Inwestorzy

Limit czasowy: 4s, limit pamięciowy: 1GB.

Jest rok 2087. Sztuczna inteligencja rozwiązała już nawet te problemy kolorowania hipergrafów, które nigdy nie zostały postawione, a ukończona sieć Hyperloop okazała się najwolniejszym metrem na świecie. Oprogramowanie opracowane w zadaniu H działało przez ostatnich 10 lat bez zarzutu, jednak jak zwykle zawiódł czynnik socjologiczny: kiedy system proponował trasy zawierające jak najdłuższe odcinki pomiędzy miastami, pasażerom spoglądającym na plan podróży ciśnienie podnosiło się do tego stopnia, że nie musieli już kupować kawy Melona po zawyżonych cenach.

Aby uspokoić nastroje wśród inwestorów, Usk postanowił zaprezentować dotychczasowe wyniki finansowe firmy. Niestety, nie były one zbyt imponujące z uwagi na niskie dochody z biletów oraz wysokie koszty zakupu kawy. Ekscentryczny miliarder postanowił więc trochę je „podrasować”.

Wyniki firmy Melona można przedstawić jako ciąg n liczb oznaczających przychód w kolejnych miesiącach. Aby nie wzbudzać podejrzeń, Usk postanowił zwiększyć liczby na pewnym spójnym fragmencie ciągu o pewną dodatnią liczbę naturalną. Wykonanie tej operacji raz nie spełniło oczekiwań wizjonera. Spróbował więc drugi raz, i trzeci... Ostatecznie uznał, że chciałby wykonać nie więcej niż k operacji podnoszenia finansów (być może za każdym razem o inną wartość).

Usk wie, że inwestorzy spoglądają przede wszystkim na wzrost przychodów firmy, a nie lubią, gdy przychody te spadają względem poprzednio osiągniętych wyników. Zależy mu więc na zminimalizowaniu liczby inwersji [1] wśród jego wyników finansowych. Czy potrafisz uratować przyszłość jego firmy?

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 400$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajdują się dwie liczby całkowite n, k ($1 \leqslant n \leqslant 6000$, $0 \leqslant k \leqslant n$).

W drugiej linii zestawu znajduje się ciąg liczb $a_1, a_2, ..., a_n$ ($0 \leqslant a_i \leqslant 10^9$), oznaczających wyniki finansowe firmy Melona.

Suma wartości n we wszystkich zestawach nie przekracza 6000.

Wyjście

Dla każdego zestawu wypisz, w osobnej linii, jedną liczbę całkowitą, będącą minimalną możliwą liczbą inwersji w ciągu, po zaaplikowaniu nie więcej niż k operacji opisanych w treści zadania.

[1] Dla przypomnienia, w ciągu $a_1, a_2, ..., a_n$, inwersją nazywamy taką parę indeksów i, j, że $i < j$ oraz $a_i > a_j$.

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
2 6 1 4 5 6 2 2 1 6 2 4 5 6 2 2 1	2 0

Zadanie J: Jeden, by wszystkimi rządzić

Limit czasowy: 2s, limit pamięciowy: 1GB.

Jeden, by wszystkimi rządzić. Jeden, by wszystkie odnaleźć. Jeden, by wszystkie zgromadzić i w ciemności związać…? Doprawdy, czarnoksiężnikowi Bajtonowi nigdy nie mieściło się to w głowie. Jak Sauron mógł być tak lekkomyślny, aby dopuścić do uzależnienia swojego losu od jednego tylko pierścienia i jeszcze się tym chwalić w inskrypcji?

Bajton również czerpie moc z magicznych pierścieni, lecz by uniknąć losu Saurona, nie stawia wszystkiego na jeden niewielki element biżuterii. Przed jego Czarną Wieżą znajduje się w rzędzie n magicznych pali, a na każdym z pali jest dokładnie k kolorowych pierścieni.

Dobry czarodziej Bitalf przepowiedział, że moc kryje się w różnorodności, a Bajton ją straci, gdy każdy pal będzie jednokolorowy (lub pusty). Sam Bitalf nie kwapi się, aby ruszać pierścienie, dał się jednak uprosić, aby wyczarować dwa dodatkowe pale – jeden po prawej, drugi po lewej stronie konstrukcji (teraz więc jest $n + 2$ pali, skrajne są puste, a na pozostałych wciąż jest po k pierścieni).

Zadanie przełożenia pierścieni spadło (jak zwykle) na hobbita, Bitbo Bajtinsa. Każdego dnia Bitbo wdrapie się niezauważony na jeden z pali, a następnie przełoży najwyższy pierścień z tego pala na jeden z sąsiadujących pali. Złe oko Bajtona na szczęście z upływem lat nieco straciło czujność i nie zauważy ani drobnego hobbita, ani różnicy w pierścieniach. Sprawę jednak utrudnia fakt, że na żadnym palu nie zmieści się więcej niż k pierścieni.

Nikt nie jest pewien, czy przepowiednia Bitalfa ma sens i pierścienie da się odpowiednio przełożyć. Pomóż dzielnemu hobbitowi i ułóż plan działania! Nie powinien być on dłuższy niż milion przełożeń, żeby Bitbo miał szansę dożyć do końca opowieści.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 25$). Potem kolejno podawane są zestawy w następującej postaci:

W pierwszej linii zestawu znajdują się dwie liczby całkowite n, k ($1 \leqslant n \leqslant 50$, $1 \leqslant k \leqslant 10$).

W każdej z następnych n linii zestawu znajduje się ciąg liczb $a_{i1}, a_{i2}, ..., a_{ik}$ ($0 \leqslant a_{ij} \leqslant 10^9$), oznaczających pierścienie na i–tym palu podane w kolejności od dołu do góry. Pale oznaczone numerami 0 i $n + 1$ istnieją, lecz są początkowo puste.

Suma wartości n we wszystkich zestawach nie przekracza 50.

Wyjście

Dla każdego zestawu w pierwszej linii wypisz `TAK` lub `NIE`, w zależności od tego, czy da się pokonać Bajtona przekładając pierścienie tak, aby docelowo na każdym palu znajdowały się pierścienie co najwyżej jednego koloru.

Jeśli odpowiedź jest twierdząca, w kolejnych liniach wypisz też plan przekładania pierścieni. W pierwszej z nich podaj liczbę przełożeń p ($0 \leqslant p \leqslant 10^6$). W kolejnych p wierszach wypisz po dwie liczby całkowite a_i, b_i ($0 \leqslant a_i, b_i \leqslant n + 1, |a_i - b_i| = 1$), oznaczające przełożenie najwyższego pierścienia z pala a_i na samą górę pala b_i. Aby wykonać tę operację, pal a_i nie może być pusty, zaś na palu b_i musi znajdować się mniej niż k pierścieni.

Jeśli istnieje wiele możliwych planów, możesz wypisać dowolny z nich, pod warunkiem zmieszczenia się w limicie 10^6 przełożeń.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
2 2 2 1 2 2 1 1 4 1 2 3 4	TAK 2 1 0 2 1 NIE

Zadanie K: Kółko i Krzyżyk

Limit czasowy: 6s, limit pamięciowy: 1GB.

Każdy gracz w „Kółko i Krzyżyk" prędzej czy później orientuje się, że istnieje prosta strategia gwarantująca remis, co czyni tę grę nieco mniej interesującą. W tym zadaniu, zamiast wykonywać optymalne ruchy w klasycznej grze w „Kółko i Krzyżyk", rozważymy jej ogólniejszy wariant i rozwiążemy problem odwrotny: czy dla danego stanu planszy możliwe jest, żeby właśnie tak zakończyła się poprawna rozgrywka?

Na potrzeby zadania przyjmujemy, że gra rozgrywana jest na planszy $n \times n$, złożonej z (początkowo pustych) kwadratów jednostkowych. Gracze naprzemiennie wykonują ruchy, w każdym ruchu wpisując swój symbol (dla jednego z graczy jest to kółko, dla drugiego krzyżyk) w wybrane puste pole. **Pierwszy ruch może zostać wykonany zarówno przez gracza grającego kółkiem jak i krzyżykiem.** Jeśli po ruchu pewnego gracza na sąsiednich polach w jednym z czterech możliwych kierunków (poziomo, pionowo, po przekątnej, lub po przeciwprzekątnej [1]) pojawi się ciąg k takich samych symboli, to gracz ten wygrywa i rozgrywka się kończy. Wreszcie, jeśli żaden z graczy nie wygra i cała plansza zostanie zapełniona, to gra kończy się remisem.

Na wejściu dana jest plansza zawierająca pewną ilość kółek, krzyżyków i wolnych pól. Rozstrzygnij, czy istnieje poprawna rozgrywka, która mogłaby zakończyć się podaną planszą. Zauważ, że w tej hipotetycznej rozgrywce gracze nie muszą grać optymalnie, a jedynie zgodnie z opisanymi wyżej zasadami gry.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 10\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, k ($3 \leqslant n \leqslant 6$, $2 \leqslant k \leqslant n$) – rozmiar planszy i liczbę sąsiednich symboli potrzebną do wygranej.

Każda z kolejnych n linii zawiera n znaków, opisujących rozważaną planszę. Możliwe znaki w opisie planszy to `.` (puste pole), `x` (krzyżyk) i `o` (kółko).

Wyjście

Dla każdego zestawu danych w pierwszej linii wypisz `TAK`, jeśli podana plansza może być finalną pozycją w poprawnej grze w „Kółko i Krzyżyk", lub `NIE`, w przeciwnym przypadku.

Jeśli odpowiedź jest twierdząca, w kolejnych liniach wypisz opis przykładowej gry, podając możliwą kolejność wypełniania pól. W i-tej linii opisu gry wypisz dwie liczby całkowite x_i, y_i ($1 \leqslant x_i, y_i \leqslant n$), oznaczające, że pole na przecięciu x_i-tego wiersza (od góry) i y_i-tej kolumny (od lewej) zostało wypełnione w i-tym ruchu. Jeśli istnieje wiele poprawnych odpowiedzi, możesz wypisać dowolną z nich.

[1] Kierunek „przekątna" oznacza kierunek skośny w prawo i dół, zaś „przeciwprzekątna" (lub antyprzekątna) to kierunek skośny w lewo i dół.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
7 3 3 x.o xxx o.o 4 3 xx.x ...o ..o. .o.. 3 3 xoo oxx xoo 3 2 xoo oxx xoo 3 3 xox .o. xox 3 2 xo. ..x xo. 3 3 x.. .x. ..x	TAK 2 2 1 3 1 1 3 3 2 1 3 1 2 3 TAK 1 1 3 3 1 2 4 2 1 4 2 4 TAK 1 2 1 1 1 3 3 1 2 1 2 2 3 2 2 3 3 3 NIE NIE NIE NIE

Wyjaśnienie

Test 1: Grę wygrał krzyżyk.

Test 2: Kółko wygrało, zajmując trzy kolejne pola w kierunku przeciwprzekątnej.

Test 3: Gra zakończyła się remisem.

Test 4: Dowolna sekwencja ruchów która mogłaby uzyskać zadaną planszę doprowadziłaby do wygranej któregoś z graczy wcześniej niż w ruchu dziewiątym.

Test 5: Krzyżyk musiał wykonać ostatni ruch, co jest sprzeczne z faktem, że wygrało kółko.

Test 6: Plansza jest poprawnym stanem gry, ale nie jest stanem końcowym.

Test 7: Krzyżyk wykonał trzy ruchy, podczas gdy kółko nie wykonało żadnego.

Zadanie L: Linie zastępcze

Limit czasowy: 10s, limit pamięciowy: 1GB.

Ordnungrad to miasto, którego mieszkańcy szczycą się utrzymywaniem idealnego porządku we wszystkich dziedzinach życia – ulice są zawsze czyste, tramwaje nadjeżdżają zgodnie z rozkładami jazdy, a komendę policji zlikwidowano po tym, gdy możliwość prowadzenia działalności przestępczej na terenie miasta została zniesiona ustawą. To właśnie ten ostatni fakt przykuł uwagę złomiarza Bajtazara.

Miejska sieć tramwajowa składa się z n skrzyżowań, przy p spośród nich zlokalizowane są pętle, zaś skrzyżowania połączone są $n-1$ dwukierunkowymi torowiskami tak, że możliwe jest przedostanie się z każdego skrzyżowania do każdego innego na dokładnie jeden sposób. Bajtazar wytypował k słabo oświetlonych torowisk i każdej z kolejnych k nocy planuje rozmontować jedno z nich, aby wywieźć pozyskany w ten sposób złom poza granice Ordnungradu.

Rządzący miastem na pewno będą woleli udawać, że torowiska są zamykane z powodu remontów, niż przyznać, że do tak poważnego zaburzenia porządku doszło za ich kadencji... a przynajmniej na to liczy Bajtazar. Ordnungradianie są jednak bardzo przywiązani do tego, że jeśli ich ulicą przez lata wykonywanych było 10 kursów tramwajowych na godzinę (łącznie na wszystkich liniach), to będzie ich właśnie 10 – ani mniej, ani więcej. Gdyby pewnego ranka władzom miasta nie udało się utworzyć siatki linii zastępczych utrzymującej niezmienione liczby kursów (na czynnych torowiskach), dla wszystkich stałoby się oczywiste, że zamknięcia wcale nie były zaplanowane! Tego Bajtazar wolałby uniknąć.

Mówiąc bardziej precyzyjnie, j-tego dnia (dla każdego $j \leqslant k$) powinno być możliwe utworzenie siatki linii zastępczych spełniającej następujące warunki:

- Każda linia musi rozpoczynać i kończyć swoją trasę na pętlach tramwajowych, wykonując pomiędzy nimi pewną liczbę kursów na godzinę (taką samą w obie strony), zaś trasa linii musi być *ścieżką prostą* [1], ponieważ tramwaje nie potrafią zawracać.
- Linie nie mogą prowadzić po torowiskach, które Bajtazar ukradł od 1-szej do j-tej nocy.
- Jeśli i-te torowisko nie zostało ukradzione, to danego dnia muszą być po nim poprowadzone linie wykonujące **w sumie** c_i kursów na godzinę (w każdą stronę).

Oczywiście, żaden miejscowy programista nie byłby skłonny pomóc Bajtazarowi, dlatego zwrócił się on właśnie do Ciebie. Znajdź poprawną kolejność kradzieży torowisk albo stwierdź, że taka kolejność nie istnieje. Zwróć uwagę, że siatka linii na każdy dzień może być ustalana niezależnie od pozostałych, musi jedynie spełniać opisane warunki.

Uwaga: Możliwe jest, że zmierzone przez Bajtazara wartości liczb c_i nie opisują poprawnej siatki linii (każdy może przecież pomylić się przy liczeniu przejeżdżających tramwajów) – w takiej sytuacji koniecznie powiadom Bajtazara o jego błędzie. Innymi słowy, opisane powyżej warunki musisz sprawdzić również w dniu $j = 0$, czyli przed dokonaniem pierwszej kradzieży.

[1] Oznacza to, że jeżeli trasa danej linii prowadzi torowiskiem ze skrzyżowania u do skrzyżowania v (i nie kończy się w v), to następnie musi ona prowadzić torowiskiem biegnącym z v do skrzyżowania innego niż u.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z ($1 \leqslant z \leqslant 15\,000$). Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, p ($2 \leqslant p \leqslant n \leqslant 500\,000$), oznaczające odpowiednio liczbę skrzyżowań oraz liczbę pętli tramwajowych.

Skrzyżowania numerowane są od 1 do n, natomiast torowiska od 1 do $n-1$.

Kolejna linia zawiera p różnych liczb całkowitych p_i ($1 \leqslant p_i \leqslant n$) uporządkowanych rosnąco, opisujących numery skrzyżowań, przy których zlokalizowane są pętle. Możesz założyć, że jeśli ze skrzyżowania wychodzi tylko jedno torowisko, to na pewno znajduje się przy nim pętla (pętle mogą być również zlokalizowane przy innych skrzyżowaniach).

Następnych $n-1$ linii opisuje kolejne torowiska. Każda z nich zawiera trzy liczby całkowite u_i, v_i, c_i ($1 \leqslant u_i, v_i \leqslant n$, $u_i \neq v_i$, $1 \leqslant c_i \leqslant 10^9$) oznaczające, że torowisko numer i łączy skrzyżowania u_i i v_i, a według pomiarów Bajtazara kursuje nim w ciągu godziny c_i tramwajów w każdą stronę.

Następna linia zestawu zawiera jedną liczbę całkowitą k ($1 \leqslant k \leqslant n-1$).

Ostatnia linia zawiera k różnych liczb całkowitych k_i ($1 \leqslant k_i \leqslant n-1$) uporządkowanych rosnąco, oznaczających numery torowisk, które Bajtazar chciałby ukraść.

Suma wartości n we wszystkich zestawach nie przekracza $2\,000\,000$.

Wyjście

Dla każdego zestawu danych, jeśli istnieje kolejność kradzieży torowisk zgodna z opisanymi warunkami, w pierwszej linii wyjścia wypisz słowo `TAK`. W kolejnej linii wypisz k liczb całkowitych r_i ($1 \leqslant r_i \leqslant n-1$), gdzie r_i jest identyfikatorem torowiska, które Bajtazar powinien rozmontować i-tej nocy. Numer każdego z k upatrzonych przez Bajtazara torowisk powinien pojawić się na tej liście dokładnie raz. Jeżeli istnieje więcej niż jedna poprawna odpowiedź, możesz wypisać dowolną z nich.

Jeśli szukana kolejność nie istnieje, lub **jeśli podane na wejściu wartości c_i nie pozwalają utworzyć poprawnej siatki linii** nawet przed rozpoczęciem kradzieży [2], wypisz słowo `NIE`.

[2] W takiej sytuacji, Bajtazar będzie musiał powtórzyć swój zwiad w celu zebrania poprawnych danych dotyczących częstotliwości kursowania tramwajów.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
2 7 5 1 2 3 4 6 7 1 3 7 2 4 7 3 4 7 4 3 5 3 1 5 6 1 2 2 3 7 5 1 2 3 4 6 7 1 3 7 2 4 7 3 4 7 4 3 5 3 1 5 6 1 4 2 3 5 6	TAK 2 3 NIE

Wyjaśnienie

W pierwszym zestawie danych, kradzież najpierw torowiska `2`, a potem torowiska `3`, pozwoli władzom miasta na opracowanie poprawnych sieci linii zastępczych po każdej zmianie:

- Przed pierwszą kradzieżą, założenia zadania spełnia między innymi następująca siatka: linia biegnąca przez skrzyżowania 1-7-2 wykonująca 3 kursy [na godzinę, w każdą stronę], linia 2-7-3-5-6 wykonująca 1 kurs oraz linia 3-7-4 wykonująca 3 kursy.
- Po kradzieży torowiska numer 2 (łączącego wierzchołki 2 i 7), możliwe jest opracowanie następującej siatki linii zastępczych: linia 1-7-3 wykonująca 1 kurs, linia 1-7-3-5-6 wykonująca 1 kurs, linia 1-7-4 wykonująca 1 kurs, linia 3-7-4 wykonująca 2 kursy.
- Po kradzieży torowisk numer 2 (łączącego wierzchołki 2 i 7) i numer 3 (łączącego wierzchołki 3 i 7), możliwe jest opracowanie następującej siatki linii zastępczych: linia 1-7-4 wykonująca 3 kursy, linia 3-5-6 wykonująca 1 kurs.

Odwrotna kolejność (`3 2`) również jest poprawną odpowiedzią.

W drugim zestawie, Bajtazar chciałby ukraść również torowiska 5 i 6. Niezależnie od tego, które z tych dwóch torowisk rozmontowałby wcześniej, po tej kradzieży nie byłoby możliwe opracowanie siatki linii poprawnie obsługującej drugie z torowisk. W związku z tym odpowiedzią jest `NIE`.

Zadanie M: Mniejsze zło

Limit czasowy: 15s, limit pamięciowy: 1GB.

Zło to zło. Mniejsze, większe, średnie, wszystko jedno. Proporcje są umowne, a granice zatarte… ale nie dla Gebajta. Dogłębne zastanawianie się nad dwuznacznościami moralnymi nie jest łatwe w stanie permanentnej glątwy [1], gdy więc pewne dwie osoby proszą Gebajta o zabicie drugiej z nich, ten bez chwili zastanowienia wie, który wybór jest mniejszym złem.

Jako potężna czarownica widzisz, że każdego z kolejnych k dni Gebajt spotka na swoim szlaku pewne dwie skonfliktowane ze sobą osoby. Wiesz też, którą z nich w każdej sytuacji zdecyduje się on zabić. Postanowiłaś wykorzystać tę wiedzę na własny użytek i doprowadzić do tego, aby kilka nielubianych przez Ciebie osób znalazło się wśród ofiar Wiedźmaka.

Twoja magia nie jest niestety dość potężna, abyś mogła zmusić Gebajta do podjęcia decyzji, którą uważa on za większe zło. Obeznana jesteś za to w urokach, co glątwę zdolne są uleczyć. Jeżeli przed jakimś spotkaniem rzucisz taki urok, to Gebajt – odzyskawszy nagle niezwykłą jasność rozumowania – zacznie się głębiej zastanawiać nad postawionym przed nim dylematem i w efekcie nie zabije tego dnia ani jednej, ani drugiej osoby. (Oczywiście, do karczmy wioskowej pod wieczór wstąpić nie omieszka i nazajutrz będzie już na powrót zmożony glątwą.)

Możesz rzucać uroki dowolnie wiele razy. Czy jesteś w stanie doprowadzić do tego, aby wszyscy Twoi nieprzyjaciele zginęli z ręki Gebajta?

Zwróć uwagę, że osoby w konfliktach mogą się powtarzać. **Konflikt ma miejsce tylko, jeżeli obie uczestniczące w nim osoby jeszcze żyją** – w przeciwnym przypadku, tego dnia nic się nie dzieje.

Wejście

Pierwsza linia wejścia zawiera liczbę zestawów danych z $(1 \leqslant z \leqslant 1\,000)$. Potem kolejno podawane są zestawy w następującej postaci:

Pierwsza linia zestawu zawiera dwie liczby całkowite n, k $(2 \leqslant n \leqslant 10^6,\ 1 \leqslant k \leqslant 10^6)$, oznaczające odpowiednio liczbę osób oraz liczbę spotkań na szlaku Gebajta. Osoby numerowane są od 1 do n.

Następnych k linii opisuje kolejne spotkania. W i-tej spośród tych linii znajdują się dwie liczby całkowite a_i, b_i $(1 \leqslant a_i, b_i \leqslant n,\ a_i \neq b_i)$. Oznaczają one, że w i-tym dniu Gebajt natrafi na konflikt pomiędzy osobami a_i i b_i, oraz zdecyduje że zabicie osoby b_i jest mniejszym złem.

Kolejna linia zawiera jedną liczbę całkowitą s $(1 \leqslant s \leqslant n)$, oznaczającą liczbę Twoich nieprzyjaciół.

Ostatnia linia zestawu zawiera s różnych liczb całkowitych s_j $(1 \leqslant s_j \leqslant n)$ uporządkowanych rosnąco. Wszystkie osoby wymienione na tej liście powinny zginąć z ręki Gebajta. Natomiast pozostałe osoby mogą, ale **nie muszą**, przeżyć.

Sumy wartości liczb n oraz k we wszystkich zestawach nie przekraczają $4\,000\,000$ każda.

Uwaga

Jeżeli Gebajt kilkukrotnie spotyka tę samą parę osób, to **nie** możesz zakładać, że jego wybór jest za każdym razem taki sam. Być może w międzyczasie zdobył on jakieś nowe informacje, które zmieniły jego osąd sytuacji. Innymi słowy, możliwe jest, że na wejściu $\exists_{i,j}(a_i = b_j \wedge a_j = b_i)$.

[1] Czyli kaca. Wcześniejsze przygody zmożonego glątwą Gebajta można odnaleźć w zadaniu „Glątwa Gebajta" z zawodów AMPPZ 2021.

Wyjście

Dla każdego zestawu w pierwszej linii wypisz pojedynczy napis `TAK` lub `NIE`, oznaczający czy dla tego zestawu danych istnieje rozwiązanie.

Jeżeli odpowiedź jest twierdząca, w kolejnej linii wypisz napis złożony z k liter `T` lub `N`. Jeżeli i-tego dnia chcesz pozwolić, aby Gebajt zabił osobę b_i, to i-tą literą wypisanego ciągu powinno być `T`. Jeśli natomiast chcesz, aby i-tego dnia Gebajt nikogo nie zabił, to i-tą literą ciągu powinno być `N`. (Jeżeli w Twoim rozwiązaniu Gebajt już wcześniej zabił osobę a_i lub osobę b_i, to nie ma znaczenia czy na i-tej pozycji wypiszesz `T` czy `N` – konflikt nie będzie mieć miejsca, a Gebajt tego dnia nie zabije nikogo.) Wypisana odpowiedź zostanie zaakceptowana, jeżeli nikt spośród s Twoich wymienionych na wejściu nieprzyjaciół nie pozostanie przy życiu.

Przykład

Dla danych wejściowych:	Możliwą poprawną odpowiedzią jest:
2 5 6 1 2 2 1 2 5 2 3 2 4 4 2 3 1 2 3 3 2 1 2 2 3 2 2 3	TAK NTTTNT NIE

Wyjaśnienie

W pierwszym zestawie danych, gdybyśmy pozwolili Gebajtowi za każdym razem wybierać mniejsze zło, to pierwszego dnia zabiłby on osobę numer 2, przez co wszystkie kolejne konflikty nie doszłyby do skutku. W związku z tym szlak Gebajta przeżyłyby osoby 1, 3, 4 oraz 5.

Jeżeli natomiast powstrzymamy Gebajta pierwszego i piątego dnia (rozwiązanie `NTTTNT`), to drugiego dnia Gebajt zabije osobę 1, trzeciego dnia osobę 5, czwartego dnia osobę 3, zaś szóstego dnia osobę 2. Przeżyje więc tylko osoba 4. Innym poprawnym rozwiązaniem jest `NTNTNT`, pozostawiające przy życiu osoby 4 oraz 5.

W drugim zestawie danych nie da się sprawić, aby zginęły zarówno osoba 2 jak i osoba 3.