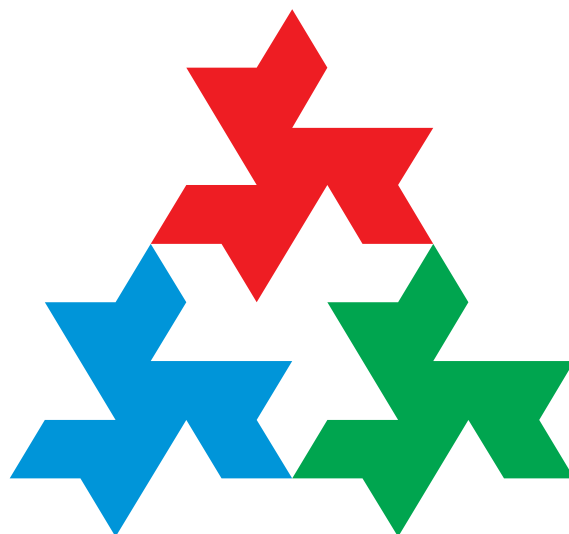


MINISTERSTWO EDUKACJI NARODOWEJ
FUNDACJA ROZWOJU INFORMATYKI
KOMITET GŁÓWNY OLIMPIADY INFORMATYCZNEJ



XXIII OLIMPIADA INFORMATYCZNA
2015/2016
TREŚCI ZADAŃ

WARSZAWA, 2016

Spis treści

Zawody I stopnia	3
<i>Hydrorozgrywka</i>	4
<i>Korale</i>	6
<i>Nadajniki</i>	9
<i>Nim z utrudnieniem</i>	12
<i>Park wodny</i>	14
Zawody II stopnia	17
<i>Świąteczny łańcuch</i>	18
<i>Drogi zmiennokierunkowe</i>	20
<i>Zająknięcia</i>	22
<i>Arkanoid</i>	24
<i>Wcale nie Nim</i>	26
Zawody III stopnia	29
<i>Równoważne programy</i>	30
<i>Posłaniec</i>	32
<i>Pracowity Jaś</i>	34
<i>Żywopłot</i>	36
<i>Klubowicze</i>	38
<i>Niebanalne podróże</i>	40
<i>Parada</i>	43
Literatura	45

Zawody I stopnia

Hydrorozgrywka

Ekipy hydrauliczne Bajtazara i Bajtoniego wygrały przetarg na doprowadzenie wody do Bajtołów Dolnych. Sieć drogowa w tym mieście składa się z n skrzyżowań połączonych m odcinkami dróg. Z każdego skrzyżowania można dojechać do każdego innego skrzyżowania za pomocą sieci dróg. Pod każdym odcinkiem drogi należy zakopać rurę wodociągową.

Aby urozmaicić sobie pracę, Bajtazar i Bajtoni postanowili zagrać w grę. Na początek ekipy obu bohaterów stają na jednym ze skrzyżowań. Przez całą grę obie ekipy będą podążać razem. Gracze wykonują ruchy na przemian, poczynwszy od Bajtazara. W swoim ruchu gracz wskazuje swojej ekipie odcinek drogi (pod którym jeszcze nie ma rury) wychodzący ze skrzyżowania, na którym znajdują się obie ekipy. Ekipa gracza zakopuje rurę pod tym odcinkiem drogi i następnie obie ekipy przemieszczają się do drugiego ze skrzyżowań, które łączy ten odcinek.

*Gracz, który nie może wykonać ruchu, przegrywa i za karę jego ekipa musi zakopać rury pod pozostałymi odcinkami dróg. Bajtazar zastanawia się, od którego skrzyżowania może zacząć się gra, aby był w stanie wygrać niezależnie od ruchów Bajtoniego. Poprosił Cię o pomoc w ustaleniu listy takich skrzyżowań. Dodatkowo zauważył, że sieć drogowa w Bajtołach ma ciekawą własność: **wyjeżdżając ze środka dowolnego odcinka drogi na dokładnie jeden sposób możemy zrobić „pętlę” i wrócić do punktu wyjścia, jeśli nigdy nie zawracamy i nie odwiedzamy żadnego skrzyżowania dwukrotnie.***

Wejście

Pierwszy wiersz standardowego wejścia zawiera dwie liczby całkowite n i m oddzielone pojedynczym odstępem, oznaczające liczbę skrzyżowań i liczbę odcinków dróg. Skrzyżowania numerujemy liczbami od 1 do n . Kolejne m wierszy opisuje sieć drogową: każdy z nich zawiera dwie liczby całkowite a, b ($1 \leq a, b \leq n$, $a \neq b$) oddzielone pojedynczym odstępem, oznaczające, że skrzyżowania o numerach a i b są połączone odcinkiem drogi. Możesz założyć, że żadne dwa skrzyżowania nie są połączone więcej niż jednym odcinkiem drogi.

Wyjście

Na standardowe wyjście należy wypisać dokładnie n wierszy: i -ty z nich ma zawierać liczbę 1, jeśli Bajtazar może wygrać, gdy gra zacznie się ze skrzyżowania numer i ; w przeciwnym wypadku ma zawierać liczbę 2.

Przykład

Dla danych wejściowych:

6 7
1 2
2 3
3 1
3 4
4 5
5 6
6 3

poprawnym wynikiem jest:

1
1
1
2
1
2

Testy „ocen”:

1ocen: $n = 9$, $m = 12$, sieć drogowa składa się z czterech „pętli”: 1–2–3–1, 1–4–5–1, 1–6–7–1 oraz 1–8–9–1.

2ocen: $n = 998$, $m = 999$, dla każdego j takiego, że $1 \leq j < n$, istnieje odcinek drogi pomiędzy j -tym i $(j + 1)$ -wszym skrzyżowaniem; istnieją również odcinki drogi łączące skrzyżowania 1 z 499 oraz 499 z 998.

3ocen: $n = 500\,000$, $m = 500\,000$, sieć dróg tworzy jedną „pętlę”.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$3 \leq n, m \leq 20$	21
2	$3 \leq n, m \leq 1000$	39
3	$3 \leq n, m \leq 500\,000$	40

Korale

Bajtyna ma n korali ponumerowanych liczbami od 1 do n . Korale są parami różne. Pewne z nich są bardziej wartościowe od innych – dla każdego z korali znana jest jego wartość w bajtalarach.

Bajtyna chciałaby stworzyć naszyjnik z niektórych ze swoich korali. Jest wiele sposobów utworzenia takiego naszyjnika. Powiemy, że dwa sposoby są różne, jeśli zbiory korali użytych do ich konstrukcji są różne. Aby nieco ułatwić sobie wybór, Bajtyna postanowiła uporządkować wszystkie sposoby utworzenia naszyjnika.

Najważniejszym kryterium jest suma wartości korali w naszyjniku. Im większa suma, tym sposób powinien być późniejszy w uporządkowaniu. Jeśli zaś mamy dwa różne sposoby utworzenia naszyjnika, które mają równą sumę wartości, to porównujemy je według porządku leksykograficznego posortowanych rosnąco list numerów użytych korali¹.

Dla przykładu rozważmy sytuację, w której są cztery korale warte kolejno (zgodnie z numeracją) 3, 7, 4 i 3 bajtalary. Z takich korali naszyjnik można utworzyć na 16 sposobów. Poniżej znajduje się uporządkowanie tych sposobów zgodnie z pomysłem Bajtyny.

Numer sposobu	Wartości wybranych korali	Suma wartości wybranych korali	Numer y wybranych korali
1	brak	0	brak
2	3	3	1
3	3	3	4
4	4	4	3
5	3 3	6	1 4
6	3 4	7	1 3
7	7	7	2
8	4 3	7	3 4
9	3 7	10	1 2
10	3 4 3	10	1 3 4
11	7 3	10	2 4
12	7 4	11	2 3
13	3 7 3	13	1 2 4
14	3 7 4	14	1 2 3
15	7 4 3	14	2 3 4
16	3 7 4 3	17	1 2 3 4

Bajtyna chciałaby stworzyć naszyjnik, który ma k -ty numer w uporządkowaniu. Pomóż jej!

¹ Ciąg numerów korali $i_1, \dots, i_p$ jest mniejszy leksykograficznie od ciągu numerów korali $j_1, \dots, j_q$, jeśli albo pierwszy ciąg jest początkowym fragmentem drugiego (czyli $p < q$, $i_1 = j_1, \dots, i_p = j_p$), albo na pierwszej pozycji, na której ciągi te różnią się, pierwszy ciąg ma mniejszy element niż drugi (czyli istnieje takie $u \in \{1, \dots, \min(p, q)\}$, że $i_1 = j_1, \dots, i_{u-1} = j_{u-1}$ oraz $i_u < j_u$).

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie dodatnie liczby całkowite n i k oddzielone pojedynczym odstępem, określające liczbę koralu oraz żądany numer sposobu utworzenia naszyjnika według uporządkowania opisanego powyżej. W drugim wierszu wejścia znajduje się ciąg n dodatnich liczb całkowitych $a_1, a_2, \dots, a_n$ pooddzielanych pojedynczymi odstępami – wartości kolejnych koralu.

Możesz założyć, że Bajtyna nie pomyliła się i rzeczywiście istnieje co najmniej k różnych sposobów utworzenia jej naszyjnika.

Wyjście

W pierwszym wierszu standardowego wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą sumę wartości koralu w znalezionym rozwiązaniu. W drugim wierszu wyjścia należy wypisać ciąg numerów koralu użytych w naszyjniku w kolejności rosnącej, rozdzielając liczby pojedynczymi odstępami.

Przykład

Dla danych wejściowych:

4 10

3 7 4 3

poprawnym wynikiem jest:

10

1 3 4

Testy „ocen”:

1ocen: $n = 10$, wszystkie korale mają wartość 1,

2ocen: $n = 9$, wartości koralu są kolejnymi potęgami dwójki,

3ocen: $n = 11$, jest jeden koral wart 1 bajtalar oraz 10 koralu wartych 10^9 bajtalarów, zaś poprawne rozwiązanie używa wszystkich jedenastu koralu.

4ocen: $n = 1\,000\,000$, $k = 10$, wartości kolejnych koralu są kolejnymi liczbami od 1 do $1\,000\,000$.

Ocenianie

Zestaw testów dzieli się na podane poniżej podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby grup testów.

We wszystkich podzadaniach zachodzą warunki $n, k \leq 1\,000\,000$ oraz $a_i \leq 10^9$.

Jeśli odpowiedź dla danego testu nie jest prawidłowa, jednak pierwszy wiersz wyjścia (suma wartości koralu w znalezionym rozwiązaniu) jest prawidłowy, wówczas przyznaje się połowę liczby punktów za dany test (oczywiście odpowiednio przeskalowaną w przypadku przekroczenia przez program połowy limitu czasowego). Dzieje się tak, nawet jeśli drugi wiersz wyjścia jest nieprawidłowy, nie został wypisany lub gdy wypisano więcej niż dwa wiersze wyjścia.

8 *Korale*

Podzadanie	Dodatkowe warunki	Liczba punktów
1	$n \leq 20, k \leq 500\,000$	8
2	$n \leq 60, k \leq 50\,000$	12
3	$n \leq 3\,000, n \cdot k \leq 10^6, a_i \leq 100$	14
4	$n \cdot k \leq 10^6$	16
5	$n \cdot k \leq 10^7$	20
6	<i>brak</i>	30

Nadajniki

Bajtazar został nowym dyrektorem zabytkowej kopalni soli pod Bajtowem. Aby zwiększyć popularność tego obiektu wśród turystów, postanowił zainstalować w korytarzach kopalni bezprzewodowy Internet.

Kopalnia składa się z n komór połączonych $n - 1$ korytarzami. Z każdej komory można przejść do każdej innej, używając korytarzy. Bajtazar postanowił rozmieścić w komorach nadajniki wi-fi tak, by Internet był dostępny w każdym z korytarzy kopalni. Aby można było korzystać z Internetu w korytarzu łączącym komory a i b , musi być spełniony co najmniej jeden z poniższych warunków:

- w komorze a lub w komorze b znajduje się nadajnik, lub
- w zbiorze komór, do których można dojść z komory a lub komory b , używając co najwyżej jednego korytarza, znajdują się co najmniej dwa nadajniki.

Bajtazar zastanawia się teraz, jaka jest minimalna liczba nadajników wi-fi, które musi rozmieścić, aby można było korzystać z Internetu w każdym korytarzu. W każdej komorze można umieścić dowolną liczbę nadajników.

Wejście

Pierwszy wiersz standardowego wejścia zawiera dodatnią liczbę całkowitą n oznaczającą liczbę komór w kopalni. Komory numerujemy liczbami od 1 do n .

Kolejne $n - 1$ wierszy opisuje korytarze w kopalni. Każdy z nich zawiera dwie liczby całkowite a i b ($1 \leq a, b \leq n$, $a \neq b$) oddzielone pojedynczym odstępem, oznaczające, że komory o numerach a i b są połączone korytarzem.

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę całkowitą, oznaczającą minimalną liczbę nadajników, które musi rozmieścić Bajtazar.

10 Nadajniki

Przykład

Dla danych wejściowych:

7

1 2

2 3

2 4

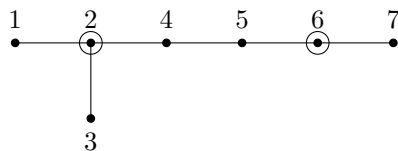
4 5

5 6

6 7

poprawnym wynikiem jest:

2



i dla danych wejściowych:

7

1 2

2 3

4 3

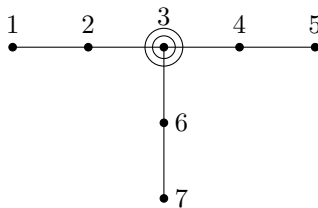
5 4

6 3

7 6

poprawnym wynikiem jest:

2



Wyjaśnienie do przykładów: W pierwszym przykładzie wystarczy umieścić nadajniki w komorach o numerach 2 i 6, natomiast w drugim przykładzie wystarczy umieścić dwa nadajniki w komorze numer 3.

Testy „ocen”:

1ocen: $n = 16$. Komora i jest połączona z komorą $\lfloor i/2 \rfloor$ dla $2 \leq i \leq n$.

2ocen: $n = 303$. Komora 2 jest połączona z komorami 1 oraz 3. Każda z komór 1, 2, 3 jest dodatkowo połączona z setką komór. Optymalnym rozwiązaniem jest umieszczenie dwóch nadajników w komorze 2.

3ocen: $n = 200\,000$. Komory i oraz $i + 1$ są połączone korytarzem dla $1 \leq i \leq n - 1$.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 10$	15
2	$n \leq 500$	20
3	$n \leq 200\,000$, do każdej komory prowadzą co najwyżej trzy korytarze	25
4	$n \leq 200\,000$	40

Nim z utrudnieniem

Ulubioną rozrywką Alicji i Bajtazara jest gra Nim. Do gry potrzebne są żetony, podzielone na kilka stosów. Dwaj gracze na przemian zabierają żetony ze stosów – ten, na którego przypada kolej, wybiera dowolny stos i usuwa z niego dowolną dodatnią liczbę żetonów. Gracz, który nie może wykonać ruchu, przegrywa¹.

Alicja zaproponowała Bajtazarowi kolejną partycjkę Nima. Aby jednak tym razem uczynić grę ciekawszą, gracze umówili się między sobą na dodatkowe warunki. Żetony, których było m , Alicja podzieliła na n stosów o licznosciach $a_1, a_2, \dots, a_n$. Zanim rozpocznie się rozgrywka, Bajtazar może wskazać niektóre spośród stosów, które zostaną natychmiast usunięte z gry. Liczba usuniętych stosów musi być jednak podzielna przez pewną ustaloną liczbę d , a ponadto Bajtazar nie może usunąć wszystkich stosów. Potem rozgrywka będzie toczyć się już normalnie, a rozpocznie ją Alicja.

Niech k oznacza liczbę sposobów, na które Bajtazar może wskazać stosy do usunięcia tak, aby mieć pewność, że wygra partię niezależnie od posunięć Alicji. Twoim zadaniem jest podanie reszty z dzielenia k przez $10^9 + 7$.

Wejście

Pierwszy wiersz standardowego wejścia zawiera dwie dodatnie liczby całkowite n i d oddzielone pojedynczym odstępem, oznaczające odpowiednio liczbę stosów i ograniczenie „podzielnościowe” zabieranych stosów.

Drugi wiersz opisuje stosy i zawiera ciąg n dodatnich liczb całkowitych $a_1, a_2, \dots, a_n$ pooddzielanych pojedynczymi odstępami, gdzie a_i oznacza liczbę żetonów na i -tym stosie.

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę całkowitą, równą liczbie sposobów (modulo $10^9 + 7$), na które Bajtazar może usunąć stosy tak, aby później na pewno zwyciężyć.

Przykład

Dla danych wejściowych:

5 2

1 3 4 1 2

poprawnym wynikiem jest:

2

Wyjaśnienie do przykładu: Bajtazar może zabrać 2 lub 4 stosy. Wygra tylko wtedy, gdy zabierze stosy o licznosciach 1 i 4 (może to zrobić na dwa sposoby).

¹ W Internecie łatwo znaleźć więcej informacji na temat gry Nim, a w szczególności opis strategii wygrywającej w tej grze.

Testy „ocen”:

1ocen: $n = 9$, $d = 2$, wynikiem jest 0,

2ocen: $n = 12$, $d = 4$,

3ocen: $n = 30$, $d = 10$, wszystkie stosy mają po 30 żetonów,

4ocen: $n = 500\,000$, $d = 2$, wszystkie stosy mają wysokość 1.

Ocenianie

Zestaw testów dzieli się na podane poniżej podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

We wszystkich podzadaniach zachodzą warunki $n \leq 500\,000$, $d \leq 10$, $a_i \leq 1\,000\,000$. Ponadto sumaryczna liczba żetonów $m = a_1 + a_2 + \dots + a_n$ jest nie większa niż $10\,000\,000$. Zwróć uwagę, że limit pamięci jest różny dla różnych podzadań.

Podzadanie	Dodatkowe warunki	Limit pamięci	Liczba punktów
1	$n \leq 20$, $a_1, \dots, a_n \leq 1000$	256 MB	10
2	$n \leq 10\,000$, $a_1, \dots, a_n \leq 1000$	256 MB	18
3	$d \leq 2$	256 MB	25
4	brak	256 MB	27
5	brak	64 MB	20

Park wodny

Park wodny Aquabajt bierze udział w konkursie na największy basen. Teren parku, na którym zlokalizowane są baseny, ma kształt kwadratu o boku długości n i jest podzielony na n^2 segmentów, z których każdy jest kwadratem o boku długości 1. Każdy z segmentów może być basenikiem albo alejką między basenikami. Baseniki połączone bezpośrednio ze sobą (czyli będące segmentami stykającymi się bokami) tworzą większe baseny. Obecnie w parku wodnym każdy basen ma kształt prostokąta.

Dyrekcja Aquabajtu postanowiła zwiększyć swoje szanse na wygraną w konkursie, przebudowując park. Ze względu na ograniczony czas i fundusze zdecydowano o przekształceniu co najwyżej dwóch segmentów z alejkami na baseniki. Pomóż władzom parku uzyskać basen złożony z maksymalnej liczby baseników. Zakładamy, że po przebudowie największy basen nie musi być już prostokątem.

Wejście

Pierwszy wiersz standardowego wejścia zawiera jedną dodatnią liczbę całkowitą n oznaczającą wielkość parku wodnego.

W następnych n wierszach znajduje się dwuwymiarowa mapa parku: każdy z tych wierszy zawiera słowo złożone z n liter. Litera A oznacza segment z alejką, natomiast litera B oznacza basenik. Możesz założyć, że w opisie znajduje się co najmniej jedna litera B.

Wyjście

Pierwszy i jedyny wiersz standardowego wyjścia powinien zawierać jedną liczbę całkowitą oznaczającą wielkość największego basenu, jaki można uzyskać.

Przykład

Dla danych wejściowych:

```
8
1 2
2 3
4 2
5 2
6 5
5 7
7 8
```

poprawnym wynikiem jest:

```
5
```

Testy „ocen”:

1ocen: $n = 10$, tylko jeden basen na planszy,

2ocen: $n = 10$, cała plansza pokryta basenem,

3ocen: $n = 1000$, szachownica.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 10$	11
2	$n \leq 50$, liczba basenów na początku nie przekroczy 80	11
3	$n \leq 60$	22
4	$n \leq 1000$, na początku każdy basen jest prostokątem 1×1	22
5	$n \leq 1000$	34

Zawody II stopnia

Świąteczny łańcuch

Każdego roku na święta Bożego Narodzenia Bajtazar dekoruje swój dom łańcuchem złożonym z różnokolorowych lampek. Tym razem Bajtazar zamierza samemu dobrać kolory lampek, które będą wchodziły w skład łańcucha. Bajtazar ma w głowie pewne wymagania estetyczne, które streszczają się w tym, że pewne fragmenty łańcucha powinny mieć identyczny układ lampek jak inne. Ponadto żona Bajtazara poprosiła go, aby tegoroczny łańcuch był jak najbardziej urozmaicony, co Bajtazar rozumie tak, że powinno w nim być jak najwięcej różnych kolorów lampek. Pomóż naszemu bohaterowi stwierdzić, ile kolorów lampek będzie musiał kupić.

Wejście

Pierwszy wiersz standardowego wejścia zawiera dwie liczby całkowite n oraz m ($n \geq 2$, $m \geq 1$) oddzielone pojedynczym odstępem, określające liczbę lampek w planowanym łańcuchu i liczbę wymagań estetycznych Bajtazara. Zakładamy, że kolejne lampki łańcucha będą ponumerowane od 1 do n . Każdy z m kolejnych wierszy opisuje jedno z wymagań za pomocą trzech liczb całkowitych a_i , b_i i l_i ($1 \leq a_i, b_i, l_i$; $a_i \neq b_i$; $a_i, b_i \leq n - l_i + 1$) oddzielonych pojedynczymi odstępami. Taki opis oznacza, że fragmenty łańcucha złożone z lampek o numerach $\{a_i, \dots, a_i + l_i - 1\}$ oraz $\{b_i, \dots, b_i + l_i - 1\}$ powinny być jednakowe. Innymi słowy, lampki o numerach a_i oraz b_i powinny mieć taki sam kolor, podobnie lampki o numerach $a_i + 1$ oraz $b_i + 1$, i tak dalej aż do lampek o numerach $a_i + l_i - 1$ i $b_i + l_i - 1$.

Wyjście

Twój program powinien wypisać na standardowe wyjście jedną dodatnią liczbę całkowitą k oznaczającą maksymalną liczbę różnych kolorów lampek, jakie mogą wystąpić w łańcuchu spełniającym wymagania estetyczne opisane na wejściu.

Przykład

Dla danych wejściowych:

10 3
1 6 3
5 7 4
3 8 1

poprawnym wynikiem jest:

3

natomiast dla danych wejściowych:

4 2
1 2 2
2 3 2

poprawnym wynikiem jest:

1

Wyjaśnienie do pierwszego przykładu: Niech a , b i c oznaczać trzy różne kolory lampek. Przykładowy łańcuch spełniający wymagania Bajtazara i jego żony to **abacbababa**.

Testy „ocen”:

1ocen: $n = 2000$, $m = 2$; Bajtazar wymaga, aby fragmenty $\{1, \dots, 1000\}$ i $\{1001, \dots, 2000\}$ były równe oraz aby fragmenty $\{1, \dots, 500\}$ i $\{501, \dots, 1000\}$ były równe; w łańcuchu może wystąpić maksymalnie 500 kolorów lampek.

2ocen: $n = 500\,000$, $m = 499\,900$; i -te wymaganie jest postaci $a_i = i$, $b_i = i + 100$, $l_i = 1$; w łańcuchu może wystąpić maksymalnie 100 kolorów lampek.

3ocen: $n = 80\,000$, $m = 79\,995$, i -te wymaganie jest postaci $a_i = i$, $b_i = i + 2$, $l_i = 4$; w łańcuchu mogą wystąpić maksymalnie dwa kolory lampek.

4ocen: $n = 500\,000$, $m = 250\,000$, i -te wymaganie jest postaci $a_i = 1$, $b_i = i + 1$, $l_i = i$; łańcuch może składać się jedynie z lampek o tym samym kolorze.

Ocenianie

Zestaw testów dzieli się na podzadania spełniające poniższe warunki. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n, m \leq 2000$	30
2	$n, m \leq 500\,000$, wszystkie liczby l_i są równe 1	20
3	$n, m \leq 80\,000$	30
4	$n, m \leq 500\,000$	20

Drogi zmiennokierunkowe

Bajtazar zastanawia się nad przeprowadzką do Bajtowa i chce wynająć tam mieszkanie. Bajtów jest pięknym miastem o licznych zaletach, choć niestety nie należy do nich komunikacja. W mieście jest n skrzyżowań połączonych mniej lub bardziej chaotyczną siecią m dróg. Drogi są bardzo wąskie, więc z przyczyn obiektywnych wszystkie są jednokierunkowe. Jakiś czas temu miejscy specjaliści od komunikacji wpadli na pomysłowe rozwiązanie, które bez konieczności poszerzania dróg umożliwia poruszanie się po nich w różnych kierunkach. A mianowicie, codziennie na wszystkich ulicach zmienia się kierunek poruszania. Innymi słowy, w dni nieparzyste ruch odbywa się zgodnie z oryginalnym skierowaniem ulic, natomiast w dni parzyste ruch na wszystkich ulicach odbywa się w przeciwnych kierunkach.

Bajtazar chce wynająć mieszkanie w takim miejscu, z którego będzie mógł wszędzie łatwo dojechać. Konkretnie, interesuje go mieszkanie przy takim skrzyżowaniu, z którego da się dojechać do każdego innego skrzyżowania **w ciągu jednego dnia** – w przypadku niektórych skrzyżowań może to być tylko nieparzysty dzień, a w przypadku innych tylko parzysty. Drogą powrotną nie trzeba się przejmować, Bajtazar może wrócić do siebie następnego dnia.

Mając daną sieć drogową w Bajtowie, wyznacz wszystkie skrzyżowania, które spełniają wymagania Bajtazara.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite n i m ($n \geq 2$, $m \geq 1$) oddzielone pojedynczym odstępem, oznaczające liczbę skrzyżowań i liczbę dróg w Bajtowie. Miasta numerujemy od 1 do n . W kolejnych m wierszach zawarte są opisy dróg: i -ty z tych wierszy zawiera dwie liczby całkowite a_i i b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) oddzielone pojedynczym odstępem, oznaczające, że istnieje jednokierunkowa droga ze skrzyżowania o numerze a_i do skrzyżowania o numerze b_i (tzn. w dni nieparzyste można przejechać tą drogą z a_i do b_i , natomiast w dni parzyste można nią przejechać z b_i do a_i). Każda uporządkowana para (a_i, b_i) wystąpi na wejściu co najwyżej raz.

Wyjście

W pierwszym wierszu standardowego wyjścia należy zapisać jedną liczbę całkowitą k oznaczającą liczbę skrzyżowań spełniających wymagania Bajtazara. W drugim wierszu należy zapisać rosnący ciąg k liczb pooddzielanych pojedynczymi odstępami, oznaczających numery tych skrzyżowań. Jeśli $k = 0$, drugi wiersz powinien pozostać pusty (tj. program może wypisać pusty wiersz albo po prostu go nie wypisywać).

Przykład

Dla danych wejściowych:

6 7

1 2

1 3

2 4

3 4

4 5

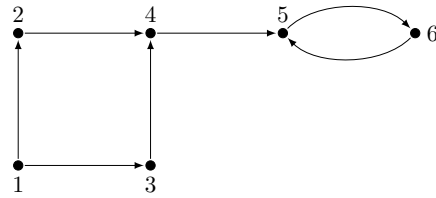
5 6

6 5

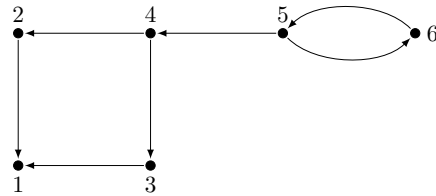
poprawnym wynikiem jest:

4

1 4 5 6



Sieć dróg w dni nieparzyste.



Sieć dróg w dni parzyste.

Wyjaśnienie do przykładu: Ze skrzyżowania numer 1 można dojechać do wszystkich innych skrzyżowań w dni nieparzyste. Ze skrzyżowań numer 5 i 6 można dojechać do wszystkich innych skrzyżowań w dni parzyste. Ze skrzyżowania numer 4 do skrzyżowań numer 5 i 6 można dojechać w dni nieparzyste, a do skrzyżowań numer 1, 2 i 3 – w dni parzyste.

Testy „ocen”:

1ocen: $n = 10$, $m = 9$, „ścieżka”, w której co druga droga jest skierowana w lewo, a co druga w prawo. Żadne skrzyżowanie nie spełnia wymagań Bajtazara.

2ocen: $n = 100\,000$, $m = 100\,000$, w dni nieparzyste można przejechać bezpośrednią drogą ze skrzyżowania numer 1 do każdego innego skrzyżowania. Dodatkowo, w dni nieparzyste można przejechać bezpośrednią drogą ze skrzyżowania numer n do skrzyżowania numer 1. Tylko skrzyżowania numer 1 i n spełniają wymagania Bajtazara.

3ocen: $n = 500\,000$, $m = 499\,999$, „ścieżka”; wszystkie skrzyżowania spełniają wymagania Bajtazara.

Ocenianie

Zestaw testów dzieli się na podzadania spełniające poniższe warunki. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n, m \leq 5000$	28
2	$n \leq 300\,000$, $m \leq 1\,000\,000$; ze wszystkich skrzyżowań spełniających wymagania Bajtazara można dojechać do każdego innego w dzień nieparzysty	29
3	$n \leq 500\,000$, $m \leq 1\,000\,000$	43

Zająknięcia

Bitek zapadł ostatnio na dziwną chorobę: strasznie się jąka, a przy tym jedyne słowa, które wypowiada, to liczby. Jego starszy brat, Bajtek, zauważył jednak dziwną powtarzalność w zająknięciach Bitka. Podejrzewa, że Bitek tak naprawdę udaje, żeby nie chodzić do szkoły i móc więcej grać na komputerze. Bajtek nie może przez to uczyć się programowania i jest z tego powodu bardzo smutny. Postanowił więc zdemaskować młodszego brata i liczy, że w nagrodę będzie miał tyle czasu na programowanie, ile dusza zapragnie.

Opiszmy formalnie podejrzenia Bajtka. Załóżmy, że mamy dany ciąg liczb A .

- **Podciągiem** A nazywamy ciąg powstały przez wyrzucenie z A dowolnych wyrazów, np. $1, 1, 7, 5$ jest podciągiem ciągu $1, 3, 1, 7, 6, 6, 5, 5$.
- **Zająknięciem** A nazywamy podciąg A , który składa się z ustawionych po kolei par takich samych wyrazów, np. $1, 1, 1, 1, 3, 3$ jest zająknięciem ciągu $1, 2, 1, 2, 1, 2, 1, 3, 3$.

Mając dane dwie wypowiedzi Bitka jako ciągi liczb, pomóż Bajtkowi stwierdzić, jaka jest długość najdłuższego zająknięcia, które występuje w każdym z tych ciągów, a nagroda Cię nie ominie.

Wejście

Pierwszy wiersz standardowego wejścia zawiera dwie liczby całkowite n oraz m ($n, m \geq 2$) oddzielone pojedynczym odstępem, oznaczające długości ciągów A i B , które reprezentują wypowiedzi Bitka. W drugim wierszu wejścia znajduje się n liczb całkowitych $a_1, a_2, \dots, a_n$ oddzielonych pojedynczymi odstępami, czyli kolejne wyrazy ciągu A ($1 \leq a_i \leq 10^9$). W trzecim wierszu wejścia znajduje się m liczb całkowitych $b_1, b_2, \dots, b_m$ oddzielonych pojedynczymi odstępami, czyli kolejne wyrazy ciągu B ($1 \leq b_i \leq 10^9$).

Wyjście

Twój program powinien wypisać na standardowe wyjście jedną nieujemną liczbę całkowitą oznaczającą długość najdłuższego wspólnego zająknięcia ciągów A i B . Jeśli ciągi nie mają żadnego wspólnego zająknięcia, poprawnym wynikiem jest 0.

Przykład

Dla danych wejściowych:

7 9

1 2 2 3 1 1 1

2 4 2 3 1 2 4 1 1

poprawnym wynikiem jest:

4

Wyjaśnienie do przykładu: Szukanym ciągiem jest $2, 2, 1, 1$.

Testy „ocen”:

1ocen: $n = 5$, $m = 4$, wszystkie liczby to 42,

2ocen: $n = 9$, $m = 13$, ciągi to słowa OLIMPIADA i INFORMATYCZNA zapisane w kodzie ASCII,

3ocen: $n = 15\,000$, $m = 15\,000$, ciąg A składa się z par rosnących liczb $(1, 1, 2, 2, 3, 3, \dots, 7500, 7500)$, natomiast B powstał w wyniku odwrócenia A ,

4ocen: $n = 10\,000$, $m = 5000$, oba ciągi składają się z par naprzemiennych liczb 13 oraz 37 $(13, 37, 13, 37, \dots)$.

Ocenianie

Zestaw testów dzieli się na podzadania spełniające poniższe warunki. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n, m \leq 2000$	30
2	$n, m \leq 15\,000$ i każda liczba w każdym ciągu występuje co najwyżej dwa razy	28
3	$n, m \leq 15\,000$	42

Arkanoid

Arkanoid jest grą komputerową, w której za pomocą ruchomej paletki odbija się poruszającą się po planszy piłeczkę. Piłeczka ta zbija znajdujące się na planszy klocki, a celem gry jest zabicie ich wszystkich. Ci, którzy grali w tę grę, wiedzą, jak frustrujące i czasochłonne może być zabicie kilku ostatnich klocków. Warto mieć zatem program, który dla początkowego ustawienia planszy obliczy czas potrzebny na wygraną gry. Na potrzeby tego zadania zakładamy dla uproszczenia, że gracz gra bezbłędnie, tzn. że zawsze odbije piłeczkę i uczyni to środkiem paletki.

Plansza ma długość m i wysokość n , przy czym m jest nieparzyste, a m i n są względnie pierwsze¹. Wprowadzamy na niej prostokątny układ współrzędnych: lewy dolny róg planszy ma współrzędne $(0, 0)$, a prawy górny współrzędne (m, n) . Dla uproszczenia zakładamy, że piłeczka ma pomijalny rozmiar, a paletka pomijalną grubość. Paletka porusza się po prostej $y = 0$, natomiast początkowo piłeczka znajduje się w punkcie $(\frac{m}{2}, 0)$ i jej początkowy wektor prędkości to $(-\frac{1}{2}, \frac{1}{2})$.

W momencie, w którym piłeczka dotknie paletki, brzegu planszy lub dowolnego klocka na planszy, odbija się idealnie sprężysto. Dodatkowo, dotknięty klocek zostaje zбитy i znika z planszy. Po ilu jednostkach czasu wszystkie klocki zostaną zбитe?

Wejście

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite m , n i k ($m, n, k \geq 1$, $k \leq nm - 1$) oddzielone pojedynczymi odstępami, oznaczające wymiary planszy oraz początkową liczbę klocków na planszy. W kolejnych k wierszach znajdują się opisy klocków: i -ty z tych wierszy zawiera dwie liczby całkowite x_i i y_i ($1 \leq x_i \leq m$, $1 \leq y_i \leq n$) oddzielone pojedynczym odstępem, oznaczające, że na planszy znajduje się klocek, który jest prostokątem o przeciwnych wierzchołkach $(x_i - 1, y_i - 1)$ oraz (x_i, y_i) . Możesz założyć, że na polu opisanym przez $x_i = \frac{m+1}{2}$, $y_i = 1$ nie znajduje się żaden klocek.

Wyjście

W jedynym wierszu standardowego wyjścia należy wypisać jedną liczbę całkowitą oznaczającą liczbę jednostek czasu, po których wszystkie klocki na planszy zostaną zбитe.

Przykład

Dla danych wejściowych:

5 4 3
2 3
5 2
3 3

poprawnym wynikiem jest:

22

¹Dwie liczby całkowite dodatnie są względnie pierwsze, jeśli ich największym wspólnym dzielnikiem jest 1.

Wcale nie Nim

Bajtoni i jego młodszy braciszek Bajtuś często grają w grę Nim. Bajtoni objaśnił braciszkowi, jaka jest strategia wygrywająca w tej grze, ale Bajtuś jeszcze nie radzi sobie z jej stosowaniem, i często przegrywa. Z tego powodu co rusz proponuje zmiany w regułach gry, mając nadzieję, że rozgrywka będzie łatwiejsza.

Właśnie zaproponował nową wersję: mamy n par stosów, przy czym stosy z i -tej pary zawierają początkowo po a_i kamieni. Gracze wykonują ruchy na przemian. Bajtuś w swoim ruchu zabiera niezerową liczbę kamieni z dowolnego wybranego przez siebie stosu. Z kolei Bajtoni w swoim ruchu przekłada niezerową liczbę kamieni pomiędzy stosami w wybranej przez niego parze. Bajtuś wykonuje ruch jako pierwszy. Przegrywa ten, kto nie może wykonać już żadnego ruchu.

Bajtoni od razu zauważył, że przy takich zasadach nie ma żadnych szans na zwycięstwo, ale nie chcąc robić przykrości braciszkowi, zgodził się zagrać. Postawił sobie jednak za punkt honoru jak najdłużej odwlekać nieuchronną przegraną. Pomóż mu i napisz program, który stwierdzi, jak długo może potrwać rozgrywka, jeśli obaj bracia grają optymalnie (Bajtuś dąży do zwycięstwa w najmniejszej liczbie ruchów, zaś Bajtoni dąży do maksymalnego wydłużenia gry).

Wejście

W pierwszym wierszu standardowego wejścia znajduje się dodatnia liczba całkowita n oznaczająca liczbę par stosów. W drugim wierszu znajduje się ciąg n dodatnich liczb całkowitych $a_1, a_2, \dots, a_n$ pooddzielanych pojedynczymi odstępami, oznaczających liczebności kolejnych par stosów.

Wyjście

W jedynym wierszu standardowego wyjścia należy zapisać jedną liczbę całkowitą oznaczającą liczbę ruchów, po których nastąpi koniec gry, jeśli obaj bracia grają optymalnie.

Przykład

Dla danych wejściowych:

2

1 2

poprawnym wynikiem jest:

7

Wyjaśnienie do przykładu: Optymalna rozgrywka może wyglądać następująco:

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

Testy „ocen”:

1ocen: $n = 1$, $a_1 = 100$, wynik 15,

2ocen: $n = 5$, wszystkie stosy po 2 kamienie, wynik 21,

3ocen: $n = 3$, $a_1 = 10^7$, $a_2 = 10^8$, $a_3 = 10^9$, wynik 163,

4ocen: $n = 3000$, $a_i = i$, wynik 65 197,

5ocen: $n = 100\ 000$, wszystkie stosy po 1 kamieniu, wynik 200 001.

Ocenianie

Zestaw testów dzieli się na podzadania spełniające następujące warunki. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów. We wszystkich podzadaniach zachodzi $a_i \leq 1\ 000\ 000\ 000$.

Podzadanie	Warunki	Liczba punktów
1	$n = 1$	10
2	suma wartości $a_i \leq 10$	10
3	$n \leq 3$	20
4	$n \leq 3\ 000$	20
5	$n \leq 500\ 000$	40

Zawody III stopnia

Równoważne programy

Bajtazar dostał nowy komputer i uczy się go programować. Program składa się z ciągu instrukcji. Jest k różnych rodzajów instrukcji, które dla uproszczenia oznaczamy liczbami od 1 do k . Niektóre pary instrukcji mają tę własność, że jeśli występują w programie bezpośrednio obok siebie (w dowolnej kolejności), to zamieniając je miejscami, nie zmienia się działania programu (czyli uzyskuje się program **równoważny**). Pozostałe pary instrukcji nie mają tej własności i nazywamy je parami **nieprzemiennymi**. Bajtazar napisał dwa programy o długości n instrukcji każdy i zastanawia się, czy są one równoważne. Pomóż mu!

Wejście

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite n , k oraz m pooddzielane pojedynczymi odstępami, oznaczające odpowiednio długość programów, liczbę różnych instrukcji komputera oraz liczbę par instrukcji nieprzemiennych.

Kolejne m wierszy zawiera opis tych par: każdy z tych wierszy zawiera dwie liczby całkowite a i b ($1 \leq a < b \leq k$) oddzielone pojedynczym odstępem, oznaczające, że para instrukcji o numerach a i b jest nieprzemienna. Możesz założyć, że każda para wystąpi w tym opisie co najwyżej raz.

Kolejne dwa wiersze przedstawiają opisy dwóch programów. Każdy z tych wierszy zawiera ciąg n liczb całkowitych $c_1, c_2, \dots, c_n$ ($1 \leq c_i \leq k$) pooddzielanych pojedynczymi odstępami, oznaczających numery kolejnych instrukcji programu.

Wyjście

W jedynym wierszu standardowego wyjścia należy wypisać jedno słowo TAK lub NIE w zależności od tego, czy podane na wejściu programy są równoważne.

Przykład

Dla danych wejściowych:

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

poprawnym wynikiem jest:

TAK

natomiast dla danych wejściowych:

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

poprawnym wynikiem jest:

NIE

Wyjaśnienie do pierwszego przykładu: W pierwszym programie można zamienić instrukcje na pozycjach 2 i 3, a następnie instrukcję na pozycji 5 z instrukcjami na pozycjach 4 i 3. W ten sposób uzyska się drugi program.

Testy „ocen”:

1ocen: $n = 50$, $k = 50$, $m = 1$; programy to $(1, 2, \dots, 49, 50)$ oraz $(50, 49, \dots, 2, 1)$; odpowiedź NIE.

2ocen: $n = 99\,999$, $k = 3$, $m = 1$; instrukcje nieprzemienne to 1 i 2, a programy to $(1, 2, 3, 1, 2, 3, \dots, 1, 2, 3)$ oraz $(3, 1, 2, 3, 1, 2, \dots, 3, 1, 2)$; odpowiedź TAK.

3ocen: $n = 100\,000$, $k = 1000$, $m = 50\,000$; programy to $(13, 13, 13, \dots, 13)$ oraz $(37, 37, 37, \dots, 37)$; odpowiedź to oczywiście NIE.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów. We wszystkich testach zachodzą warunki $1 \leq n \leq 100\,000$, $1 \leq k \leq 1000$, $0 \leq m \leq 50\,000$.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 5$	5
2	$k \leq 2$	5
3	$n \leq 1000$	25
4	brak dodatkowych warunków	65

Posłaniec

Bajtazar po długim okresie swojego panowania w królestwie Bajtocji stwierdził, że to zajęcie bardzo go wyczerpało, i ustąpił z tronu. Jednak przywykł on do życia w wyższych sferach i chciałby pozostać na bieżąco z najważniejszymi wiadomościami dotyczącymi królestwa i dworu. Dlatego postanowił zostać królewskim posłańcem.

Już pierwszego dnia na nowym stanowisku zlecono mu dostarczenie pilnej wiadomości pomiędzy dwoma miastami królestwa. Stwierdził on jednak, że w trakcie pracy zrobi sobie małą wycieczkę krajoznawczą i niekoniecznie pojedzie najkrótszą możliwą trasą. Oczywiście nie może dopuścić, aby nowy król się o tym dowiedział – w końcu posłaniec powinien dostarczać wiadomości tak szybko, jak to tylko możliwe!

Wszystkie połączenia drogowe w Bajtocji są jednokierunkowe. Bajtazar zna doskonale całe królestwo i wie, między którymi miastami istnieją połączenia drogowe. Zadeklarował on liczbę połączeń, których chciałby użyć, przejeżdżając pomiędzy miastami, i planuje on przejechać pomiędzy nimi dowolną ścieżką wymagającą dokładnie tylu połączeń (nie zważając na to, ile połączeń tak naprawdę wymaga taka podróż). W trakcie swojej podróży Bajtazar nie może jednak pojawić się w początkowym ani końcowym mieście więcej niż raz, gdyż wtedy wzbudziłby podejrzenia u królewskich oficjeli. Może on jednak wielokrotnie pojawiać się w innych miastach, jak też wielokrotnie korzystać z tych samych połączeń drogowych.

Pomóż naszemu bohaterowi i napisz program, który obliczy dla niego, na ile sposobów może on zrealizować swoją wycieczkę krajoznawczą. Innymi słowy, program ma wyznaczyć liczbę różnych ścieżek o zadanej długości pomiędzy dwoma wybranymi miastami królestwa (przy czym w mieście początkowym i końcowym można pojawić się tylko raz). Ponieważ wynik zapytania może być całkiem duży, wystarczy, że program poda resztę z dzielenia wyniku przez pewną wybraną przez Bajtazara liczbę.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się trzy liczby całkowite n , m i z ($n \geq 2$, $0 \leq m \leq n(n-1)$, $2 \leq z \leq 1\,000\,000\,000$) pooddzielane pojedynczymi odstępami, oznaczające odpowiednio: liczbę miast w Bajtocji, liczbę jednokierunkowych połączeń między nimi oraz liczbę wybraną przez Bajtazara. Miasta numerujemy liczbami od 1 do n .

Dalej następuje m wierszy; każdy zawiera parę liczb całkowitych a , b ($1 \leq a, b \leq n$, $a \neq b$) oddzielonych pojedynczym odstępem, opisującą jednokierunkowe połączenie z miasta o numerze a do miasta o numerze b . Żadne połączenie nie jest podane na wejściu wielokrotnie.

W kolejnym wierszu wejścia znajduje się dodatnia liczba całkowita q oznaczająca liczbę zapytań Bajtazara. W każdym z następnych q wierszy znajduje się opis jednego zapytania. Opis taki składa się z trzech liczb całkowitych u_i , v_i i d_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq d_i \leq 50$) pooddzielanych pojedynczymi odstępami, oznaczających, że Bajtazar ma przejechać z miasta o numerze u_i do miasta o numerze v_i , używając dokładnie d_i połączeń.

Wyjście

Na standardowe wyjście należy wypisać dokładnie q wierszy. W i -tym wierszu należy wypisać resztę z dzielenia przez z liczby ścieżek z i -tego zapytania.

Przykład

Dla danych wejściowych:

5 7 10

1 2

2 3

3 4

4 5

5 1

2 4

4 1

2

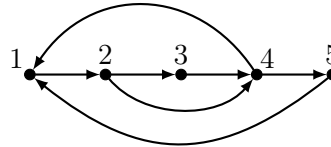
2 1 3

5 3 6

poprawnym wynikiem jest:

2

1



Wyjaśnienie do przykładu: Dla pierwszego zapytania mamy dwie możliwe ścieżki: $2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ oraz $2 \rightarrow 4 \rightarrow 5 \rightarrow 1$; dla drugiego zapytania tylko jedną: $5 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 1 \rightarrow 2 \rightarrow 3$.

Testy „ocen”:

1ocen: $n = 6$, $q = 10$, pięć miast połączonych ze sobą bezpośrednio w obie strony; szóste miasto niepołączone z żadnym innym;

2ocen: $n = 20$, $q = 100$, miasta w Bajtocji położone są na okręgu; każde dwa sąsiednie miasta na tym okręgu są połączone ze sobą bezpośrednio w obie strony;

3ocen: $n = 100$, $q = 500\,000$, mapa Bajtocji ma kształt trójkąta.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 20$, $q \leq 100$	12
2	$n \leq 100$, $m \leq 500$, $q \leq 100$	20
3	$n \leq 100$, $q \leq 10\,000$	38
4	$n \leq 100$, $q \leq 500\,000$	30

Pracowity Jaś

Jaś miał niedawno urodziny. Jak w każdym szanującym się zadaniu algorytmicznym, Jaś nie dostał w prezencie ani zabawek, ani gier, ani komputera, a jedynie długie tablice wypełnione liczbami, drzewa, mapy różnych krajów z drogami prowadzącymi tunelami i estakadami oraz długie taśmy z wypisanymi 1048576 początkowymi literami słów Fibonacciego i Thuego-Morse’a. Najbardziej z tych wszystkich prezentów spodobała mu się tablica z wypisaną permutacją¹ pierwszych n liczb naturalnych. Zaczął się zastanawiać, jaka jest poprzednia permutacja w porządku leksykograficznym². Po chwili udało mu się ją wymyślić i zapragnął zapisać ją na tej samej tablicy. Jaś potrafi w jednym kroku zamienić miejscami jedynie dwie z liczb zapisanych na tablicy (gdyby operował na większej liczbie liczb naraz, to by się pogubił). Jest jednak na tyle mądry, że przekształcił początkową permutację w poprzednią leksykograficznie, wykonując minimalną liczbę takich zamian. Gdy to zrobił, wpadł w permutacyjny szal i zaczął powtarzać tę operację w kółko, zapisując na tablicy kolejne wcześniejsze permutacje w porządku leksykograficznym!

Niestety, po pewnym czasie Jaś będzie musiał przerwać swoją zabawę, gdyż dojdzie do permutacji $1, 2, \dots, n$, która jest najmniejsza w porządku leksykograficznym. Przyjaciele Jasia trochę się naśmiewają z jego monotonnej zabawy, jednak wiedzą, że nie mają szans go z niej wyrwać. Chcieliby się więc dowiedzieć, kiedy Jaś skończy. Pomóż kolegom Jasia i powiedz im, ile potrwa jego zabawa, jeśli każda zamiana zajmuje mu dokładnie sekundę. Jako że ta rozrywka może być dość długa (Jaś nie bez powodu ma przydomek Pracowity), wystarczy im reszta z dzielenia tej liczby przez $10^9 + 7$. W końcu $10^9 + 7$ sekund to na tyle długo, że są w stanie co tyle czasu sprawdzać, czy Jaś już skończył swoją zabawę.

Wejście

W pierwszym wierszu standardowego wejścia znajduje się dodatnia liczba całkowita n oznaczająca długość permutacji, którą Jaś otrzymał na urodziny. W drugim wierszu znajduje się ciąg n parami różnych liczb naturalnych $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$) pooddzielanych pojedynczymi odstępami, opisujący tę permutację.

Wyjście

Twój program powinien wypisać na standardowe wyjście resztę z dzielenia przez $10^9 + 7$ liczby zamian, które wykona Jaś, zanim jego zabawa się skończy.

¹Permutacja liczb od 1 do n to ciąg różnych liczb całkowitych $p_1, \dots, p_n$ spełniających $1 \leq p_i \leq n$ (każda liczba całkowita od 1 do n występuje w takiej permutacji dokładnie raz).

²Permutacja $P = (p_1, \dots, p_n)$ jest wcześniej w porządku leksykograficznym niż permutacja $Q = (q_1, \dots, q_n)$ (co zapiszemy jako $P < Q$), jeżeli $p_j < q_j$, gdzie j jest najmniejszym takim indeksem, że $p_j \neq q_j$. Permutacja P poprzedza permutację Q w porządku leksykograficznym, jeżeli $P < Q$ oraz nie istnieje taka permutacja R , że $P < R < Q$.

Przykład

Dla danych wejściowych:

3

3 1 2

poprawnym wynikiem jest:

6

Wyjaśnienie do przykładu: Na tablicy Jasia będą się po kolei pokazywały permutacje $(2, 3, 1)$, $(2, 1, 3)$, $(1, 3, 2)$, $(1, 2, 3)$. Aby je uzyskiwać, będzie musiał wykonać łącznie $2 + 1 + 2 + 1 = 6$ zamian.

Testy „ocen”:

1ocen: $1, 2, 3, \dots, 10$

2ocen: losowa 5-elementowa permutacja

3ocen: $100, 99, 98, \dots, 1$.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 10$	15
2	$n \leq 5000$	37
3	$n \leq 1\,000\,000$, permutacja to $n, n-1, \dots, 1$	15
4	$n \leq 1\,000\,000$	33

Żywopłot

Królewski ogrodnik Bajtazar ma za zadanie wyhodować w królewskim ogrodzie labirynt z żywopłotu. Ogród można podzielić na $m \times n$ kwadratowych pól. Otoczony jest murem, w którym na środku północnej i południowej ściany są wejścia. Na każdej krawędzi dzielącej dwa pola można zbudować kawałek żywopłotu – z cisu lub tui. Król bardziej lubi cis, więc chciałby mieć w swoim ogrodzie jak najwięcej kawałków żywopłotu z cisu. Niestety, cis wymaga lepszej gleby, więc nie wszędzie go można posadzić.

Aby żywopłot tworzył labirynt, musi spełniać dodatkowy warunek: do każdego pola musi być możliwość dojścia z obu wejść i, co więcej, tylko na jeden sposób. (Z danego pola można przejść bezpośrednio na pole sąsiadujące, jeśli na dzielącej te pola krawędzi nie znajduje się kawałek żywopłotu. Dwa sposoby dojścia uznajemy za różne, jeśli przechodzą przez różne zbiory pól.)



W lewej części powyższego rysunku przedstawiono przykładowy ogród dla $m = 4$ i $n = 5$, zawierający 31 krawędzi. Wyróżniono w nim 13 krawędzi, na których można posadzić żywopłot z cisu.

Na prawej części rysunku przedstawiono przykładowy labirynt składający się z 12 kawałków żywopłotu, z których 10 jest żywopłotem z cisu, a 2 są żywopłotem z tui. Nie istnieje labirynt zawierający więcej kawałków z cisu. Twoim zadaniem będzie napisanie programu, który pomoże Bajtazarowi w zaprojektowaniu labiryntu.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite m i n oznaczające rozmiar ogrodu ($2 \leq m, n$ oraz n jest liczbą nieparzystą). Kolejne m wierszy zawiera po $n - 1$ znaków, opisujących pionowe krawędzie (czytane rzędami, od lewej do prawej). Znak C oznacza, że na danej krawędzi można posadzić żywopłot z cisu, a znak T oznacza, że można posadzić żywopłot z tui. Kolejne $m - 1$ wierszy zawierające po n znaków opisuje poziome krawędzie (również czytane rzędami, od lewej do prawej).

Wyjście

W pierwszym wierszu standardowego wyjścia należy wypisać dwie liczby całkowite: liczbę posadzonych kawałków żywopłotu tworzących labirynt oraz maksymalną liczbę kawałków żywopłotu

z *cisu*. W kolejnych $2m - 1$ wierszach należy opisać krawędzie labiryntu (w kolejności jak na wejściu). Należy wypisać znak Z, jeśli krawędź zawiera żywopłot, lub znak . (kropka) w przeciwnym wypadku.

Jeśli istnieje wiele rozwiązań spełniających warunki króla, należy wypisać dowolne z nich.

Przykład

Dla danych wejściowych:

4 5

CCTT

TTCT

TCTT

TTCT

CCCTT

TCCCT

CTCTT

jednym z poprawnych wyników jest:

12 10

Z..Z

..Z.

.Z.Z

..Z.

.ZZ..

.Z.Z.

Z.Z..

Wyjaśnienie do przykładu: Dane wejściowe opisują ogród z lewej części rysunku; wynik opisuje labirynt z prawej części rysunku.

Testy „ocen”:

1ocen: $m = 4$, $n = 3$, w każdym miejscu można posadzić *cis*;

2ocen: $m = 100$, $n = 99$, na pionowych krawędziach można posadzić *cis*, na poziomych można posadzić *tuję*;

3ocen: $m = 1000$, $n = 999$, w każdym miejscu można posadzić *tuję*.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Jeśli Twój program wypisze poprawny pierwszy wiersz, a dalsza część wyjścia nie będzie poprawna, uzyska 52% punktów za dany test. W szczególności, aby uzyskać 52% punktów za test, wystarczy wypisać tylko jeden wiersz wyjścia.

Podzadanie	Warunki	Liczba punktów
1	$n \cdot m \leq 12$	25
2	$n, m \leq 100$	25
3	$n, m \leq 1000$	50

Klubowicze

Bajtocki Klub Dyskusyjny jest wyjątkowy pod każdym względem. Posiada on 2^n członków, z których każdy zadeklarował, jakie ma poglądy na n fundamentalnych pytań. Konkretne sformułowanie pytań nie jest istotne, wystarczy wiedzieć, że są to pytania, na które można udzielić jednej z dwóch odpowiedzi (np. „kawa czy herbata?”). Poglądy danej osoby możemy kodować za pomocą ciągu bitów, który interpretowany w systemie binarnym da liczbę całkowitą z przedziału od 0 do $2^n - 1$.

W klubie nie ma dwóch osób o jednakowych poglądach. Powiemy, że dwie osoby są **prawie zgodne**, jeśli ich poglądy różnią się tylko na jednym pytaniu. Ponadto klubowicze to 2^{n-1} panów i 2^{n-1} pań, którzy tworzą 2^{n-1} par. Klubowicze spotykają się przy **okrągłym stole**. Chcemy ich tak usadzić, żeby każdy klubowicz siedział obok swojej partnerki lub swojego partnera, a obok siebie po drugiej stronie miał osobę prawie zgodną.

Wejście

W pierwszym wierszu standardowego wejścia znajduje się liczba całkowita n oznaczająca liczbę fundamentalnych pytań. W kolejnych 2^{n-1} wierszach znajdują się opisy par klubowiczów: w i -tym z tych wierszy znajdują się dwie liczby całkowite a_i, b_i ($0 \leq a_i, b_i \leq 2^n - 1$) oddzielone pojedynczym odstępem, oznaczające, że klubowicze o zestawie poglądów opisanym liczbami a_i i b_i są parą. Każda liczba reprezentująca klubowicza pojawi się na wejściu dokładnie raz.

Wyjście

Jeśli nie istnieje usadzenie klubowiczów spełniające warunki zadania, to w jedynym wierszu standardowego wyjścia należy wypisać jedno słowo NIE.

Jeśli takie usadzenie istnieje, to w jedynym wierszu standardowego wyjścia należy wypisać ciąg 2^n liczb całkowitych pooddzielanych pojedynczymi odstępami, oznaczający poprawne usadzenie klubowiczów przy okrągłym stole.

Jeśli istnieje wiele poprawnych odpowiedzi, należy wypisać dowolną z nich.

Przykład

Dla danych wejściowych:

3
0 5
4 1
3 6
7 2

poprawnym wynikiem jest:

0 5 7 2 6 3 1 4

Testy „ocen”:

1ocen: $n = 4$, jeśli i jest parzyste, to klubowicze o numerach i oraz $i + 1$ są w parze;

2ocen: $n = 10$, jeśli i jest nieparzyste, to klubowicze o numerach i oraz $i + 1$ są w parze;
wyjątkiem jest klubowicz $2^n - 1$, który jest w parze z klubowiczem 0;

3ocen: $n = 15$, test losowy, pary na wejściu są posortowane rosnąco względem liczb a_i .

Ocenianie

Zestaw testów dzieli się na 18 grup, z których każda warta jest 5 albo 6 punktów. W grupie numer k znajdują się wyłącznie testy z $n = k + 1$ (a zatem $2 \leq n \leq 19$).

Niebanalne podróże

Bajtazar ostatnio polknął bakcyła podróżowania po Bajtocji. W kraju tym jest n miast (które dla uproszczenia numerujemy liczbami od 1 do n), a bajtocka kolej oferuje podróżnym m dwukierunkowych połączeń kolejowych pomiędzy niektórymi parami miast. Używając tych połączeń, Bajtazar może się dostać do każdego miasta w Bajtocji (być może musi się przy tym przesiadać).

Nasz bohater szczególnie upodobał sobie podróże, w których wyrusza z pewnego miasta, by na końcu do niego wrócić, ale nie odwiedzając przy tym w trakcie podróży żadnego miasta dwukrotnie ani nie używając żadnego połączenia dwukrotnie. Takie podróże nazywa **niebanalnymi**.

W trakcie kolejnej ze swoich wielu podróży Bajtazar zauważył, że każda niebanalna podróż, w którą wyruszył, używała tyle samo połączeń kolejowych. Podejrzuje on, że jest to uniwersalna własność sieci kolejowej w Bajtocji, i poprosił Ciebie o zweryfikowanie tej hipotezy. Ponadto, jeżeli hipoteza jest prawdziwa, to chciałby on także poznać liczbę różnych niebanalnych podróży, które może odbyć. Z wiadomych sobie powodów Bajtazar zadowolony jest jedynie resztą z dzielenia liczby takich podróży przez $10^9 + 7$.

Podróż możemy formalnie opisać za pomocą ciągu liczb oznaczających kolejno odwiedzane miasta. Dwie podróże o tej samej długości są różne, jeśli istnieje taki indeks i , że i -te miasta z kolei odwiedzone w trakcie tych podróży są różne. Przez długość podróży rozumiemy liczbę połączeń, których ona używa.

Wejście

W pierwszym wierszu standardowego wejścia znajdują się dwie liczby całkowite n i m ($n \geq 1$, $m \geq 0$) oddzielone pojedynczym odstępem, oznaczające odpowiednio liczbę miast w Bajtocji oraz liczbę oferowanych połączeń. Dalej następuje m wierszy opisujących oferowane połączenia. W i -tym z tych wierszy znajdują się dwie liczby całkowite a_i i b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$) oddzielone pojedynczym odstępem, oznaczające, że istnieje dwukierunkowe połączenie kolejowe umożliwiające podróż pomiędzy miastami o numerach a_i i b_i . Między każdą parą miast biegnie co najwyżej jedno bezpośrednie połączenie kolejowe.

Wyjście

Jeżeli pechowo okazało się, że nie istnieje ani jedna niebanalna podróż, to na standardowe wyjście należy wypisać jedno słowo BRAK. Jeżeli istnieją takie podróże, ale nie wszystkie mają tę samą długość (czyli hipoteza Bajtazara jest fałszywa), należy wypisać jedno słowo NIE. W końcu jeżeli wszystkie niebanalne podróże mają taką samą długość (czyli hipoteza jest prawdziwa), to należy wypisać jedno słowo TAK, a w kolejnym wierszu dwie liczby całkowite oddzielone pojedynczym odstępem, oznaczające długość niebanalnych podróży oraz resztę z dzielenia liczby niebanalnych podróży przez $10^9 + 7$.

Przykład

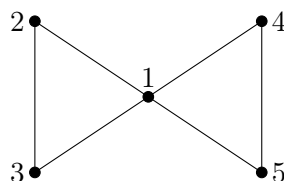
Dla danych wejściowych:

5 6
1 2
2 3
3 1
1 4
4 5
5 1

poprawnym wynikiem jest:

TAK

3 12



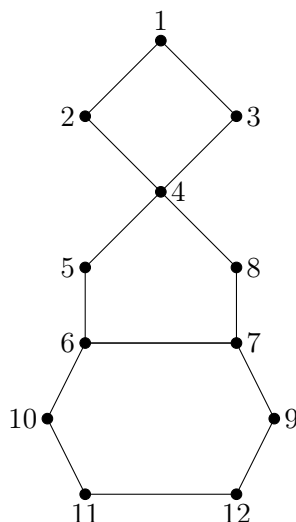
Wyjaśnienie do przykładu: Wszystkie niebanalne podróże mają długość 3 i jest ich 12. Są to kolejno 1-2-3-1, 1-3-2-1, 2-1-3-2, 2-3-1-2, 3-1-2-3, 3-2-1-3, 1-4-5-1, 1-5-4-1, 4-1-5-4, 4-5-1-4, 5-1-4-5, 5-4-1-5.

natomiast dla danych wejściowych:

12 14
1 2
2 4
3 1
4 3
4 5
5 6
6 7
7 8
8 4
7 9
9 12
12 11
11 10
10 6

poprawną odpowiedzią jest:

NIE



Testy „ocen”:

1ocen: $n = 500\,000$, miasta w Bajtoci leżą na ścieżce, odpowiedź to oczywiście BRAK.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

42 Niebanalne podróże

Podzadanie	Warunki	Liczba punktów
1	$n \leq 18$	20
2	$n, m \leq 2000$	40
3	$n \leq 500\,000, m \leq 1\,000\,000$	40

Dostępna pamięć: 64 MB.

OI, etap III, dzień drugi, 14.04.2016

Parada

Jak co roku na powitanie wiosny ulicami Bajtogradu przejdzie Wielka Wiosenna Parada Bajtowska. Swoją obecnością uświetni ją sam Król Bajtazar XVI. Sieć drogowa Bajtogradu składa się z n skrzyżowań połączonych $n-1$ dwukierunkowymi odcinkami ulic (z każdego skrzyżowania da się dojechać do każdego innego).

Dokładna trasa parady nie jest jeszcze znana, ale wiadomo, że zacznie się ona w jednym ze skrzyżowań, będzie biegła pewną liczbą odcinków ulic i zakończy się na **innym** skrzyżowaniu. Aby nie zanudzić paradujących, trasa przechodzić będzie przez każdy odcinek ulicy co najwyżej raz.

Z uwagi na bezpieczeństwo uczestników parady, należy zamknąć bramką wlot każdego odcinka ulicy, przez który nie przechodzi parada, a który wchodzi do skrzyżowania, przez które parada przechodzi (włączając początkowe i końcowe skrzyżowanie). Należy wyznaczyć, ile takich bramek może być potrzebnych.

Wejście

W pierwszym wierszu standardowego wejścia znajduje się liczba całkowita n ($n \geq 2$) oznaczająca liczbę skrzyżowań w Bajtogradzie. Skrzyżowania numerujemy liczbami od 1 do n .

Kolejne $n-1$ wierszy opisuje sieć drogową Bajtogradu. Każdy z nich zawiera dwie liczby całkowite a i b ($1 \leq a, b \leq n$, $a \neq b$) oddzielone pojedynczym odstępem, oznaczające, że skrzyżowania o numerach a i b są połączone dwukierunkowym odcinkiem ulicy.

Wyjście

W pierwszym i jedynym wierszu standardowego wyjścia należy wypisać jedną liczbę całkowitą, oznaczającą maksymalną liczbę bramek, które mogą być potrzebne do zabezpieczenia parady.

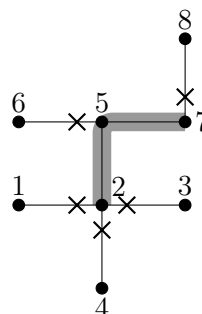
Przykład

Dla danych wejściowych:

```
8
1 2
2 3
4 2
5 2
6 5
5 7
7 8
```

poprawnym wynikiem jest:

```
5
```



44 Parada

Wyjaśnienie do przykładu: Jeśli parada ruszy ze skrzyżowania 2 i zakończy się na skrzyżowaniu 7, to potrzebne będzie 5 bramek (3 do zamknięcia wlotów do skrzyżowania 2 i po jednej do zamknięcia wlotów do skrzyżowań 5 i 7).

Testy „ocen”:

1ocen: $n = 20$, ścieżka;

2ocen: $n = 20$, gwiazda;

3ocen: $n = 1000$, losowy test o następującej własności: i -ty odcinek ulicy (dla $i = 1, \dots, n-1$) łączy skrzyżowanie numer $i+1$ z jednym ze skrzyżowań o mniejszych numerach.

Ocenianie

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby osobnych grup testów.

Podzadanie	Warunki	Liczba punktów
1	$n \leq 20$	15
2	$n \leq 300$	16
3	$n \leq 3000$	22
4	$n \leq 200\,000$	47

Bibliografia

- [1] *I Olimpiada Informatyczna 1993/1994*. Warszawa–Wrocław, 1994.
- [2] *II Olimpiada Informatyczna 1994/1995*. Warszawa–Wrocław, 1995.
- [3] *III Olimpiada Informatyczna 1995/1996*. Warszawa–Wrocław, 1996.
- [4] *IV Olimpiada Informatyczna 1996/1997*. Warszawa, 1997.
- [5] *V Olimpiada Informatyczna 1997/1998*. Warszawa, 1998.
- [6] *VI Olimpiada Informatyczna 1998/1999*. Warszawa, 1999.
- [7] *VII Olimpiada Informatyczna 1999/2000*. Warszawa, 2000.
- [8] *VIII Olimpiada Informatyczna 2000/2001*. Warszawa, 2001.
- [9] *IX Olimpiada Informatyczna 2001/2002*. Warszawa, 2002.
- [10] *X Olimpiada Informatyczna 2002/2003*. Warszawa, 2003.
- [11] *XI Olimpiada Informatyczna 2003/2004*. Warszawa, 2004.
- [12] *XII Olimpiada Informatyczna 2004/2005*. Warszawa, 2005.
- [13] *XIII Olimpiada Informatyczna 2005/2006*. Warszawa, 2006.
- [14] *XIV Olimpiada Informatyczna 2006/2007*. Warszawa, 2007.
- [15] *XV Olimpiada Informatyczna 2007/2008*. Warszawa, 2008.
- [16] *XVI Olimpiada Informatyczna 2008/2009*. Warszawa, 2009.
- [17] *XVII Olimpiada Informatyczna 2009/2010*. Warszawa, 2010.
- [18] *XVIII Olimpiada Informatyczna 2010/2011*. Warszawa, 2011.
- [19] *XIX Olimpiada Informatyczna 2011/2012*. Warszawa, 2012.
- [20] *XX Olimpiada Informatyczna 2012/2013*. Warszawa, 2013.
- [21] *XXI Olimpiada Informatyczna 2013/2014*. Warszawa, 2015.
- [22] A. V. Aho, J. E. Hopcroft, J. D. Ullman. *Projektowanie i analiza algorytmów komputerowych*. PWN, Warszawa, 1983.

- [23] L. Banachowski, A. Kreczmar. *Elementy analizy algorytmów*. WNT, Warszawa, 1982.
- [24] L. Banachowski, K. Diks, W. Rytter. *Algorytmy i struktury danych*. WNT, Warszawa, 1996.
- [25] J. Bentley. *Perelki oprogramowania*. WNT, Warszawa, 1992.
- [26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. *Wprowadzenie do algorytmów*. WNT, Warszawa, 2004.
- [27] M. de Berg, M. van Kreveld, M. Overmars. *Geometria obliczeniowa. Algorytmy i zastosowania*. WNT, Warszawa, 2007.
- [28] R. L. Graham, D. E. Knuth, O. Patashnik. *Matematyka konkretna*. PWN, Warszawa, 1996.
- [29] D. Harel. *Algorytmika. Rzecz o istocie informatyki*. WNT, Warszawa, 1992.
- [30] D. E. Knuth. *Sztuka programowania*. WNT, Warszawa, 2002.
- [31] W. Lipski. *Kombinatoryka dla programistów*. WNT, Warszawa, 2004.
- [32] E. M. Reingold, J. Nievergelt, N. Deo. *Algorytmy kombinatoryczne*. WNT, Warszawa, 1985.
- [33] K. A. Ross, C. R. B. Wright. *Matematyka dyskretna*. PWN, Warszawa, 1996.
- [34] R. Sedgewick. *Algorytmy w C++. Grafy*. RM, 2003.
- [35] S. S. Skiena, M. A. Revilla. *Wyzwania programistyczne*. WSiP, Warszawa, 2004.
- [36] P. Stańczyk. *Algorytmika praktyczna. Nie tylko dla mistrzów*. PWN, Warszawa, 2009.
- [37] M. M. Sysło. *Algorytmy*. WSiP, Warszawa, 1998.
- [38] M. M. Sysło. *Piramidy, szyszki i inne konstrukcje algorytmiczne*. WSiP, Warszawa, 1998.
- [39] J. Tomasiewicz. *Zaprzyjaj się z algorytmami. Przewodnik dla początkujących i średniozaawansowanych*. PWN, Warszawa, 2016.
- [40] R. J. Wilson. *Wprowadzenie do teorii grafów*. PWN, Warszawa, 1998.
- [41] N. Wirth. *Algorytmy + struktury danych = programy*. WNT, Warszawa, 1999.
- [42] *W poszukiwaniu wyzwań. Wybór zadań z konkursów programistycznych Uniwersytetu Warszawskiego*. Warszawa, 2012.
- [43] *W poszukiwaniu wyzwań 2. Zadania z Akademickich Mistrzostw Polski w Programowaniu Zespołowym 2011-2014*. Warszawa, 2015.