

# Osiągalność wierzchołków Kontratakuje

PREOI 2025

Dzień 0 – 24 stycznia 2025

Kod zadania: **owk**  
Limit pamięci: **64 MiB**



Dany jest graf. Chcemy wiedzieć, do jakich wierzchołków da się dojść z wierzchołka nr 0, idąc tylko po krawędziach.

Brzmi znajomo? To pa tera

Panna A. i pan K. to nierozłączna para. Robią razem niemal wszystko: jedzą posiłki, chodzą na spacer, odrabiają lekcje i określają osiągalność wierzchołków w grafie. Tym razem jednak, to właśnie w tej ostatniej czynności pojawia się problem... Jako przykładni uczniowie Staszica codziennie wyznaczają wierzchołki, do których da się dojść ze źródła. Jak u większości uczniów Staszica, jest to najistotniejsza część nie tylko poranka, ale i całego dnia! Niestety państwo A. i K., przyzwyczajeni do stwierdzania, do jakich wierzchołków można się przemieścić, zapomnieli, jak samodzielnie obliczać zbiór wierzchołków leżących w tej samej spójnej składowej co podany wierzchołek.

Co więcej, ponieważ są niesłychanie zdolną parą, dawno temu opracowali sposób, w jaki każde z nich może zapamiętać tylko część grafu. Korzystając z niezwykłej zdolności *rozumienia się bez słów*, wspólnie ustalają, czy da się dojść do każdego wierzchołka ze źródła. Wracając do naszej dramatycznej historii: nasi bohaterowie mieli właśnie zabrać się za rytualne obliczanie wiadomo czego, gdy okazało się, że pan K. zatrzaskał się w łazience.

Co gorsza, mimo szpary na dole, drzwi łazienki są całkowicie dźwiękoszczelne. Tymczasem pani Ś. zażyczyła sobie, aby problem został rozwiązany wcześniej rano, a **pani Ś. się nie odmawia**. W panice pan K. znalazł w łazience nitkę dentystyczną oraz papierowy kubeczek. Myślicie o tym samym, co ja? Otóż pan K. również tak pomyślał. Może i telefon na sznurek nie zastąpi mocy *rozumienia się bez słów*, ale w tak kryzysowej sytuacji musi wystarczyć.

Rozmowa przez kubeczek działa w następujący sposób:

- Podczas każdej sekundy jedna osoba przykładła swoją buzię do kubeczka, a druga do ucha.
- Jeśli w pewnym momencie zarówno panna A. jak i pan K. przykładają swoje uszy albo buzie do kubeczka, dochodzi do nieporozumienia, nigdy wcześniej taka sytuacja im się nie wydarzyła, więc można bezpiecznie założyć, że wtedy nie rozwiążą problemu.
- Ponieważ jakość sznurka nie jest zbyt dobra, podczas komunikacji można odróżnić tylko dźwięki wysokie od niskich, więc w każdej sekundzie są w stanie przekazać sobie dokładnie jeden bit informacji.
- Pani Ś. czeka, więc chcą dogadać się jak najszybciej.

## Formalnie

Mamy dwa procesy, każdy z nich dostaje liczbę  $n$  oraz część krawędzi z grafu. Celem programów jest wyznaczenie, do których wierzchołków można dojść zaczynając z wierzchołka o numerze 0. Ich celem jest przy tym minimalizowanie liczby wysłanych między sobą bitów.

## Implementacja

Twój program będzie uruchamiany w dwóch procesach i za każdym razem będzie wywoływany z inną funkcją:



### Uruchomienie dla Panny A.:

- `void mainA(int n, std::vector<pair<int, int>> edges)`

### Uruchomienie dla Pana K.:

- `void mainK(int n, std::vector<pair<int, int>> edges)`

Obydwie funkcje mogą wywoływać następujące funkcje:

- `void send(bool)` – ta funkcja służy do przesyłania bitów drugiej osobie.
- `bool receive()` – ta funkcja służy do czytania bitów od drugiej osoby.
- `void answer(std::vector<bool>)` – ta funkcja służy do zwrócenia wyniku. Może ją wywołać dowolny proces, po jej wywołaniu obydwa procesy zostają zatrzymane (w szczególności, drugi proces nie musi wiedzieć, że pierwszy już odpowiada).

Program **musi** zaczynać się od linijki `#include "owklib.h"`

## Ważne uwagi

Program może używać zmiennych globalnych, ale ponieważ jest on uruchamiany oddzielnie dla Panny A. i Pana K., uruchomienia te nie będą współdzielić zmiennych globalnych. Nie należy samodzielnie zakończyć wywołania programu, jedyne poprawne zakończenia to: wywołanie funkcji `answer` oraz koniec funkcji `mainA/K`.

Dla uczestników korzystających z systemu operacyjnego Linux (lub WSL itp.) – do czego bardzo gorąco zachęcamy – w zakładce pliki został umieszczony plik `dłazaw.zip` (w kategorii (owk)) znajduje się tam parę plików ale należy modyfikować jedynie plik `owk.cpp` oraz korzystać z pliku `compile.sh` (poniżej wyjaśnienie).

Dla uczestników nie mających dostępu do Linux'a udostępniamy środowisko w przeglądarce. Aby z niego skorzystać, należy wejść w link, założyć konto (jeśli takowego się jeszcze nie ma) i w prawym górnym rogu kliknąć `Use this template -> Open in codespaces`.

Do dyspozycji jest skrypt `compile.sh`, aby z niego skorzystać należy wpisać komendę w terminal: `./compile.sh 0` albo `./compile.sh 1` – odpowiednio bez/z pomocniczymi informacjami. Skrypt ten stworzy plik `owk.e`, który następnie należy odpalić (komendą `./owk.e`) i wpisać test.

Testy mają następującą strukturę:

```
n A K
a1,1 a1,2
a2,1 a2,2
...
aA,1 aA,2

k1,1 k1,2
k2,1 k2,2
...
kK,1 kK,2
```

gdzie  $n$  to liczba wierzchołków grafu,  $A$  liczba krawędzi znanych Pannie A.,  $K$  liczba krawędzi znanych Panu K..  
 $a_{i,1}, a_{i,2}$  to oczywiście krawędzie znane Pannie A. oraz  $k_{i,1}, k_{i,2}$  to krawędzie znane Panu K..

Wyjście programu będzie dwojakie, albo zacznie się od ERROR po czym zostanie wskazany konkretny błąd, albo zacznie się od ANSWER po czym nastąpi liczba będąca liczbą bitów jakie Państwo A. i K. wykorzystali do porozumienia się, a w kolejnej linii nastąpi  $n$  liczb 0 lub 1,  $i$ -ta z nich mówiąca o tym czy  $i$ -ty wierzchołek jest osiągalny z wierzchołka 0.

## Ograniczenia

Liczba wierzchołków grafu jest dodatnią liczbą całkowitą i nie przekroczy 8000, a liczba krawędzi 16000. Można bezpiecznie założyć, że w grafie nie ma multikrawędzi (powtarzających się krawędzi) oraz pętli (krawędzi wychodzących i wchodzących do tego samego wierzchołka)

## Przykłady

Wejście dla testu owk0a:

```
4 2 0
0 2
3 2
```

Wyjście dla testu owk0a:

```
1 0 1 1
```

Wejście dla testu owk0b:

```
6 3 2
3 4
2 1
0 3
4 0
0 5
```

Wyjście dla testu owk0b:

```
1 0 0 1 1 1
```

## Ocenianie

Podzadanie	Ograniczenia	Limit czasu	Liczba punktów
1	wszystkie krawędzie otrzymuje Panna A.	5 s	10
2	dany graf jest lasem	5 s	50
3	brak dodatkowych ograniczeń	5 s	40

Niech  $b$  to liczba bitów której Panna A. i Pan. K wykorzystali do porozumienia się.

Liczba punktów jakie rozwiązanie otrzyma na danym teście to

$$\begin{cases} 0 & b > 208000 \\ 100 & b < 112000 \\ 50 + \left\lfloor \frac{50(208000-b)}{208000-112000} \right\rfloor & \text{w przeciwnym wypadku} \end{cases}$$

Inaczej mówiąc wynik skaluje się liniowo między 50 a 100 punktów dla liczby bitów między 208000, a 112000.