



Bit Algo
START



Grafy - ciąg dalszy



Zadanie 1: Krasnoludy i trolle

Wyobraźmy sobie podziemny labirynt, złożony z ogromnych jaskiń połączonych wąskimi korytarzami. W jednej z jaskiń krasnoludy zbudowały swoją osadę, a w każdej z pozostałych jaskiń mieszka znana krasnoludom ilość trolli. Krasnoludy chcą zaplanować swoją obronę na wypadek ataku ze strony trolli. Zamierzają w tym celu zakraść się i podłożyć ładunek wybuchowy pod jeden z korytarzy, tak aby trolle mieszkające za tym korytarzem nie miały żadnej ścieżki, którą mogłyby dotrzeć do osady krasnoludów.

Który korytarz należy wysadzić w powietrze, aby odciąć dostęp do krasnoludzkiej osady największej liczbie trolli?



Zadanie 2: Dostarczanie przesyłek

Bajtocja jest krainą zawierającą N miast, $N-1$ dwukierunkowych dróg i układ dróg tworzy graf spójny. Mając listę K miast do których musimy dostarczyć przesyłki i mogąc wystartować i zakończyć trasę w dowolnym mieście, podaj minimalny dystans, który musimy przebyć, że zrealizować to zadanie.



Zadanie 3: Szach i goniec (II kolokwium 2020)

Algocja leży na wielkiej pustyni i składa się z miast oraz oaz połączonych drogami. Każde miasto jest otoczone murem i ma tylko dwie bramy. Z każdej bramy prowadzi dokładnie jedna droga do jednej oazy (ale do danej oazy może dochodzić dowolnie wiele dróg; oazy mogą też być połączone drogami między sobą). Prawo Algocji wymaga, że jeśli ktoś wjechał do miasta jedną bramą, to musi go opuścić drugą.

Szach Algocji postanowił wysłać gońca, który w każdym mieście kraju odczyta zakaz formułowania zadań “o szachownicy” (obraza majestatu). Szach chce, żeby goniec odwiedził każde miasto dokładnie raz (ale nie ma ograniczeń na to ile razy odwiedzi każdą z oaz). Goniec wyjeżdża ze stolicy Algocji, miasta x , i po odwiedzeniu wszystkich miast ma do niej wrócić.

Proszę przedstawić (bez implementacji) algorytm, który stwierdza czy odpowiednia trasa gońca istnieje. Proszę uzasadnić poprawność algorytmu oraz oszacować jego złożoność czasową.



Zadanie 4: Domino

Mamy pewien układ klocków domino. Otrzymujemy go w postaci listy par $[a, b]$: Jeżeli przewrócimy klocek a , to klocek b też się przewróci. Chcemy znaleźć minimalną liczbę klocków, które trzeba przewrócić ręcznie, aby wszystkie domina były przewrócone.



Zadanie 5: Ścieżki w DAGu

Otrzymujemy na wejściu w postaci listy krawędzi nieskierowany graf acykliczny (DAG - Directed Acyclic Graph) oraz parę wierzchołków s i t . Naszym zadaniem jest obliczyć, ile jest możliwych ścieżek między s i t .



Bit Algo
START