

Grafy

11 kwietnia 2019

Zarys zadania

Dane będą 3 różne reprezentacje grafów skierowanych. Dane będą 2 algorytmy. Zadaniem jest napisać kod w takiej postaci, aby algorytmy były napisane tylko raz i mogły działać niezależnie od prawdziwej reprezentacji grafu. Kod ma być rozszerzalny w tym sensie, że stworzenie nowej reprezentacji grafów nie spowoduje jakichkolwiek zmian w kodzie algorytmów.

Podpowiedź: wszystkie grafy powinny znać swoją liczbę wierzchołków.

Reprezentacje grafów

Cykl (*Cycle*)

Graf o n wierzchołkach przechowuje tablicę liczb będącą pewną permutacją liczb $0 \dots n - 1$. Np. graf o 4 wierzchołkach przechowuje tablicę $[0, 2, 1, 3]$. Oznacza to, że krawędzie (skierowane) znajdują się pomiędzy wierzchołkami: $0 \rightarrow 2, 2 \rightarrow 1, 1 \rightarrow 3, 3 \rightarrow 0$.

Siatka (*Grid*)

Siatka zna swoją wysokość h i szerokość w . Wierzchołków jest łącznie wh w (w każdym wierszu jest ich w , a w każdej kolumnie h), na planie prostokąta. Jeśli mamy 3 wierzchołki wiersz i 2 wiersze, to będą krawędzie (krawędzie idą z lewej do prawej lub z góry na dół, ale nigdy nie na odwrót): $0 \rightarrow 1, 1 \rightarrow 2, 0 \rightarrow 3, 1 \rightarrow 4, 2 \rightarrow 5, 3 \rightarrow 4, 4 \rightarrow 5$.

Graf na macierzy (*MatrixGraph*)

Najbardziej typowa reprezentacja. Jeśli graf ma n wierzchołków, to przechowuje macierz $n \times n$. Brak krawędzi reprezentujemy przez 0. Krawędź przez 1. $M[i, j] = 1$ oznacza krawędź od wierzchołka i -tego do j -tego.

Algorytmy

Wypisywanie krawędzi (*PrintEdges*)

Pierwszy algorytm wypisuje krawędzie, jakie są w danym grafie.

Czy Eulerowski? (*IsEuler*)

Drugi algorytm sprawdza, czy graf jest Eulerowski. Graf jest Eulerowski, gdy dla każdego wierzchołka jego stopień wchodzący równy jest stopniowi wychodzącemu.

Co można

Można dowolnie modyfikować dostarczone klasy reprezentujące grafy. Można tworzyć nowe klasy i interfejsy. Nie należy modyfikować funkcji `Main()`.