

05.06.2017r.

Grafi i sieci – projekt 16

1. Temat projektu

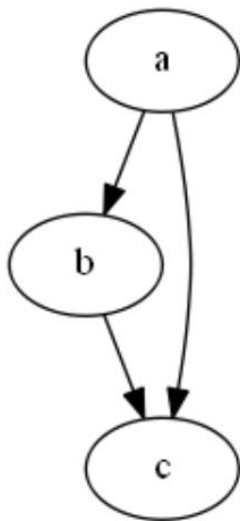
Opracować i zaimplementować algorytm sprawdzania czy graf jest dwudzielny.

2. Jak rozumiemy temat

Graf dwudzielny to taki graf, którego wierzchołki możemy podzielić na 2 podzbiory, w taki sposób, że każda krawędź grafu łączy wierzchołki z różnych podzbiorów. Innymi słowy w takim grafie można pokolorować dwoma kolorami wierzchołki w taki sposób, by wierzchołki w tym samym kolorze nie były ze sobą połączone.

3. Algorytmy

Jeśli podany graf jest grafem skierowany to zamieniamy go na graf nieskierowany, celu wykrycia wszystkich cykli. Na obrazku poniżej pokazany jest „cykl” uniemożliwiający powstanie grafu dwudzielnego, który w grafie nieskierowanym nie zostałby wykryty.



W naszym programie użyliśmy procedur `DFS()` oraz `explore(i)` omawianych na ćwiczeniach, w celu znalezienia wszystkich krawędzi drzewowych danego grafu. Następnie sprawdzamy niewykryte krawędzie w celu sprawdzenia czy w grafie nie istnieje cykl o długości nieparzystej.

Procedura `DFS()` sprawdza po kolei wszystkie wierzchołki i jeśli napotkany wierzchołek jest nieodwiedzony to wywołuje dla niego procedurę `explore(i)`, gdzie `i` to numer tego wierzchołka.

Procedura `explore(i)` zapisuje do listy wierzchołków informację, iż dany wierzchołek (`i`) jest odwiedzony, zapisuje liczbę `prev` danego wierzchołka i sprawdza wszystkich jego sąsiadów. Jeśli któryś jest nieodwiedzony to wywołuje dla niego procedurę `explore(j)`, gdzie `j` to numer tego wierzchołka.

Jeśli sprawdzeni zostaną wszyscy sąsiedzi procedura zwiększy licznik i powróci do wierzchołka wywołującego tą metodę.