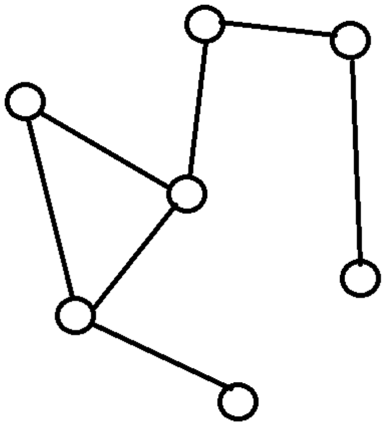
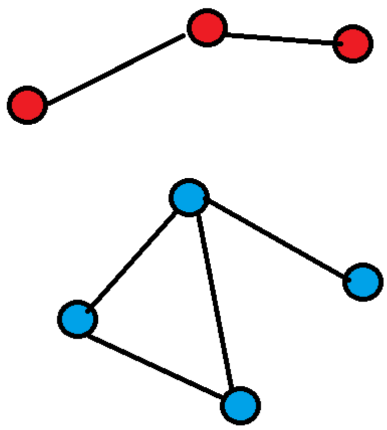
Spójność grafu oraz wyznaczanie liczby spójnych składowych grafu nieskierowanego

Graf nazywamy spójnym gdy dla każdej pary wierzchołków istnieje ścieżka, która je łączy.



Graf spójny



Graf niespójny (posiadający dwie spójne składowe)

Spójne składowe

Spójną składową grafu nazywamy największą grupę wierzchołków, grafu, które są wzajemnie połączone (spójne). Spójna składowa musi zawierać przynajmniej jeden wierzchołek 😊. Graf spójny posiada tylko jedną spójną składową (w takim wypadku liczba wierzchołków tej jedynej składowej jest równa liczbie wierzchołków całego grafu). Graf niespójny ma co najmniej dwie spójne składowe.

Jak zbadać czy graf nieskierowany jest spójny:

Sprawdzić, ile wierzchołków zwróci nam algorytm DFS lub BFS (startujący z dowolnego wierzchołka). Jeśli jest ich tyle, ile wierzchołków w całym grafie to graf jest spójny. Gdy jest ich mniej, to graf jest niespójny.

Wyznaczanie spójnych składowych grafu nieskierowanego

W tym zakresie interesują nas dwie kwestie:

1) Ile graf ma spójnych składowych
2) Jakie wierzchołki wchodzą w skład poszczególnych spójnych składowych grafu.

Na obydwa pytania można odpowiedzieć za pomocą algorytmu opartego na DFS (przy czym BFS też się do tego nadaje). Algorytm wygląda tak:

1) Tworzymy pomocniczą macierz (lub rozszerzalny wektor) X zmiennych typu int, który ma tyle elementów ile wierzchołków w grafie. Macierz tą na początku wypełniamy zerami. I-ty element tej macierzy informuje nas do jakiej spójnej składowej należy dany wierzchołek (0 oznacza, że dany wierzchołek nie został jeszcze przypisany do żadnej spójnej składowej).
2) Inicjalizujemy sobie licznik składowych spójnych na L=1.
3) Teraz mamy główną część algorytmu. Dopóki w macierzy X znajdują się jakiekolwiek zera musimy kontynuować następującą procedurę (będzie to pętla while).

   Za pomocą pętli for (która jest wewnątrz pętli while) przechodzimy przez wszystkie elementy macierz X. Jeśli podczas wykonywania pętli w jakimkolwiek elemencie macierzy X (dajmy na to i-tym) napotkamy na zero to :

   A1) Tak ustawiamy warunek pętli while z punktu 3, aby doszło do jej kolejnej iteracji (kolejne przejście przez pętle for)

   A2) Uruchamiamy algorytm DFS startujący od wspomnianego i-tego wierzchołka (tego wykrytego w pętli for, dla którego w tabeli X było 0). Dla wszystkich wierzchołków zwróconych przez algorytm DFS w tabeli X ustawiamy wartość L. Przykładowo powiedzmy, że DFS zwrócił nam m.in. wierzchołek numer n wtedy X[n]=L (oczywiście z reguły DFS zwróci nam wiele wierzchołków a nie jeden, zmienić wartość w macierzy X trzeba dla wszystkich).

   A3) Zwiększamy L o jeden (to jest bardzo ważny warunek). Po wykonaniu tych czynności można przejść do kolejnego kroku pętli for.
4) Odczyt wyników wygląda tak. Po wykonaniu pętli while aktualna wartość L pomniejszona o 1 określa nam ilość spójnych składowych. Można ją też odczytać, jako maksymalną liczbę przechowywaną w macierzy X. Aby wyznaczyć jakie wierzchołki wchodzą do poszczególnych spójnych składowych trzeba przeanalizować macierz X. Wszystkie wierzchołki przechowujące w macierzy X 1 wchodzą w skład 1 składowej. Wszystkie wierzchołki przechowujące 2 wchodząc w skład 2 składowej itd. (oczywiście numer wierzchołka jest tożsamy z indeksem elementu w macierzy).