

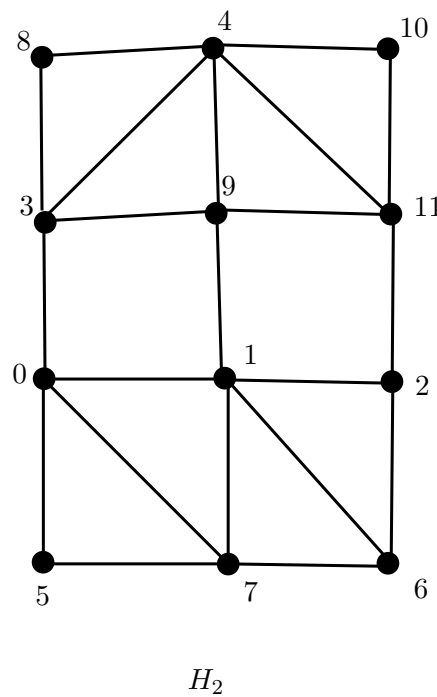
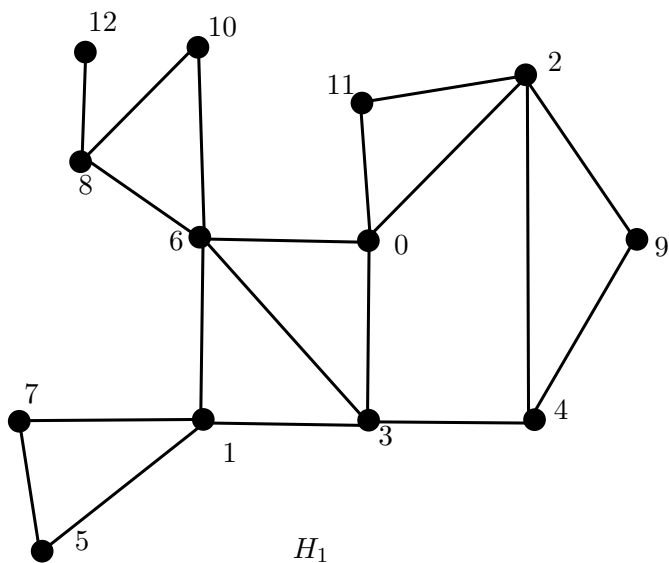
Algorytm BFS i DFS (dla grafów nieskierowanych)

1. Dla grafów H_1, H_2 wyznacz drzewo BFS (zaczynając od wierzchołka 3).
2. W grafie H_2 wyznacz odległości wszystkich wierzchołków od wierzchołka nr 5.
3. Dla grafów H_1, H_2 przedstawionych na rysunkach i dla grafu H_3 (podany poniżej) wyznacz drzewo DFS (lub las DFS, w przypadku grafów niespójnych), zaczynając od wierzchołka nr 3. Wyznacz porządki preorder, postorder.

$$H_3 = (V, E),$$

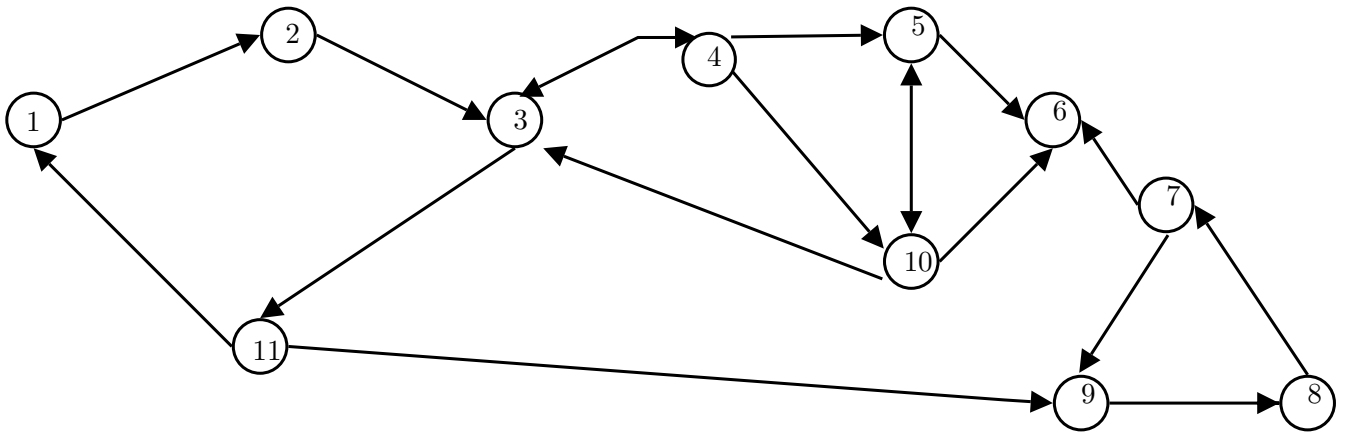
$$V = \{0, 1, \dots, 14\}, E = \{\{0, 3\}, \{0, 9\}, \{1, 2\}, \{1, 12\}, \{2, 4\}, \{2, 5\}, \{2, 6\}, \{3, 9\}, \{3, 14\}, \{4, 13\}, \{5, 13\}, \{6, 12\}, \{7, 10\}, \{7, 11\}, \{7, 12\}, \{8, 10\}, \{8, 12\}, \{9, 14\}, \{10, 11\}\}.$$

4. Niech G będzie dowolnym spójnym grafem, o co najmniej dwóch wierzchołkach. Udowodnij, że G ma wierzchołek, którego usunięcie (wraz z wszystkimi incyduentnymi krawędziami) nie prowadzi do powstania grafu niespójnego.
5. Wykorzystaj algorytm DFS do wyznaczania wierzchołków rozspajających w spójnym grafie, dla grafu H_1 (stosując algorytm Hopcroft i Tarjan).

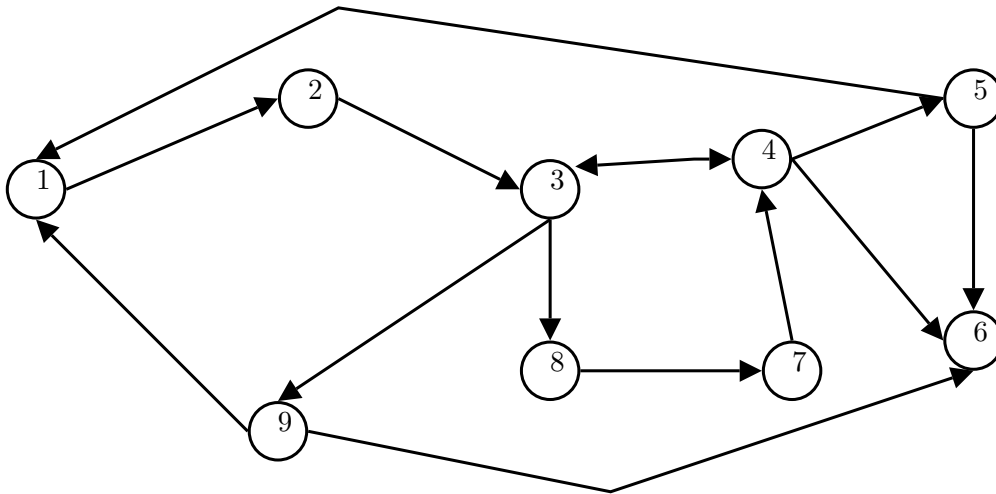


DFS i BFS dla grafów skierowanych (digrafów)

1. Dla podanych digrafów D_1, D_2 wyznacz ich DFS-drzewa (lub lasy) oraz BFS-drzewa (lasy).
2. Wskaż w digrafach D_1, D_2 cykle skierowane i silnie spójne składowe.
3. Uzasadnij, że jeżeli D' jest silnie spójną składową digrafu D , to D' jest też silnie spójną składową w D_{rev} .
4. Za pomocą algorytmu Kosaraju–Sharir wyznacz silnie spójne składowe w digrafach D_2, D_1 .
5. Za pomocą algorytmu BFS, wyznacz w digrafie D_1 silnie spójną składową zawierającą wierzchołek nr 1 (ułóż odpowiedni algorytm). Dlaczego za pomocą tego algorytmu można w czasie liniowym zdecydować, czy dany digraf jest silnie spójny? Czy można wykorzystać ten algorytm, aby wyznaczyć wszystkie silnie spójne składowe digrafu? W jakim czasie?



D_1



D_2

DFS i BFS — zadania

1. Dana jest klasa `AlgorytmBFS` implementująca algorytm BFS. Dodaj do klasy `AlgorytmBFS` metodę `int odleglosc(int v)` zwracającą odległość wierzchołka v od wierzchołka startowego. (Odległość wierzchołków u i v , ozn. $d(u, v)$, definiujemy jako długość najkrótszej $u - v$ -drogi w grafie; jeżeli wierzchołki u i v nie są połączone drogą (tzn. są w różnych składowych grafu), to ich odległość $d(u, v)$

wynosi ∞ . Można wykorzystać np. wartość `Integer.MAX_VALUE` (lub `Double.POSITIVE_INFINITY`) jako $d(u, v)$ w tym przypadku.)

- Wyznacz ekscentryczności wierzchołków w grafie H_1 , średnicę, promień i centrum tego grafu. W jaki sposób można wykorzystać algorytm BFS do wyznaczania ekscentryczności wierzchołka i pozostałych parametrów: średnica, promień i centrum grafu?

(Ekscentryczność wierzchołka u w grafie G definiujemy $e(v, G) = \max\{d(v, u) : u \in V(G)\}$,

promień grafu $r(G) = \min\{e(v, G) : v \in V(G)\}$,

średnica grafu $\text{diam}(G) = \max\{d(v, u) : v, u \in V(G)\}$.

Zbiór $\text{CT}(G) = \{v \in V(G) : e(v, G) = r(G)\}$ nazywamy centrum grafu G .)

Wyznacz powyższe parametry dla grafu H_4 , rzędu 8, danego w postaci listy krawędzi:

$\{0, 5\}, \{0, 7\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{4, 5\}, \{5, 6\}, \{6, 7\}, \{3, 4\}$.

- Utwórz klasę `GrafWlasnosc`, która będzie zawierała metody:

`int ekscent(Graf G, int u)` zwracającą ekscentryczność wierzchołka u w grafie G ;

`int radius(Graf G)` zwracającą promień grafu G ;

`int diameter(Graf G)` zwracającą średnicę grafu G ;

`ArrayList<Integer> centrum(Graf G)` zwracającą listę wierzchołków stanowiących centrum grafu.

Wykorzystaj metodę `odleglosc` z klasy `AlgorytmBFS` do wyznaczania odległości wierzchołka v od wierzchołka startowego.

Uwaga. Powyższe parametry mają sens tylko dla grafów spójnych!

Wykorzystaj klasę `GrafWlasnosc` aby wyznaczyć powyższe parametry dla grafu H_4 (najpierw zapisz graf H_4 do pliku `txt` w odpowiednim formacie).

- (1) Zmodyfikuj kod klasy `AlgorytmDFS`, aby można było wyznaczać spójne składowe grafu za pomocą algorytmu DFS: indeksujemy składowe liczbami od 0 do $k - 1$, gdzie k jest liczbą składowych i dla każdego wierzchołka v grafu zapisujemy, do której składowej należy v .

(2) Zmodyfikuj klasę `AlgorytmDFS` w taki sposób, aby umożliwiała wyznaczenie liczby wierzchołków każdej składowej spójności grafu.

- Ułóż algorytm, bazujący na DFS, który będzie określał, czy dany graf jest dwudzielny. Prześledź działanie algorytmu na małych grafach. Następnie zaproponuj implementację tego algorytmu. (Możesz uzupełnić definicję poniższej klasy `DFSdwudzielny`, dodając odpowiednie pola oraz uzupełniając kod metod.)

```
public class DFSdwudzielny{  
  
    public DFSdwudzielny(Graf G)  
    private void dfs(Graf G, int v)  
    public boolean czyDwudzielny( )  
}
```

- Ułóż algorytm, bazujący na DFS lub BFS, który będzie określał, czy dany graf prosty jest acykliczny. Prześledź działanie algorytmu na małych grafach. Następnie zaproponuj implementację tego algorytmu.

(*) Czy krawędzie wielokrotne są przeszkodą w działaniu algorytmu? A pętle? Zmodyfikuj algorytm tak, aby umożliwiał też analizę grafów zawierających pętle (krawędzie wielokrotne).