Algorytmy i Struktury Danych II, Zestaw 5

WYKRYWANIE CYKLI Z WYKORZYSTANIEM ALGORYTMU DFS

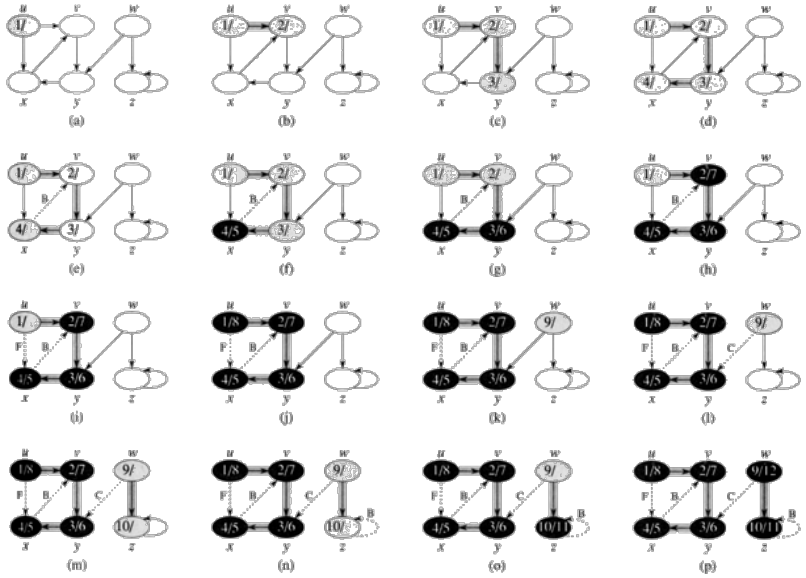
Proszę napisać program wykrywający cykle w grafie skierowanym, działający w oparciu o algorytm przeszukiwania "w głąb" (DFS - *Depth-first search*).

Działania algorytmu tłumaczy poniższy pseudokod:

```
DFS(G)
1  for each vertex u ∈ G.V
2      u.color = WHITE
3      u.π = NIL
4  time = 0
5  for each vertex u ∈ G.V
6      if u.color == WHITE
7          DFS-VISIT(G, u)

DFS-VISIT(G, u)
 1  time = time + 1            // white vertex u has just been discovered
 2  u.d = time
 3  u.color = GRAY
 4  for each v ∈ G.Adj[u]      // explore edge (u, v)
 5      if v.color == WHITE
 6          v.π = u
 7          DFS-VISIT(G, v)
 8  u.color = BLACK            // blacken u; it is finished
 9  time = time + 1
10  u.f = time
```

Na poniższym rysunku zaprezentowano przykład zastosowania algorytmu przeszukiwania grafu DFS.



W procesie przeszukiwania grafu z wykorzystaniem algorytmu DFS, każdemu wierzchołkowi przypisywana jest etykieta czasowa d/f, gdzie d to czas pierwszych odwiedzin a f czas drugich odwiedzin. Wierzchołkom przypisywane są kolory: BIAŁY (nie odwiedzony wierzchołek), SZARY (jednokrotnie odwiedzony wierzchołek), CZARNY (dwukrotnie odwiedzony wierzchołek). Ponadto, krawędzie grafu podlegają następującej klasyfikacji: T (krawędzie drzewowe) - prowadzą do nieodwiedzonego węzła, F (krawędzie w przód) - krawędzie niedrzewowe prowadzące do potomka, B (krawędzie powrotne) - krawędzie skierowane do przodka, C (krawędzie poprzeczne) - wszystkie inne krawędzie.

Istnienie cyklu w grafie jest równoważne występowaniu krawędzi typu B.

Materiały pomocnicze: Link 1, Link 2.

Zachęcam do poczytania na temat generowania labiryntów z wykorzystaniem algorytmu DFS: Link 3.