

## ALGORYTMY DO ZESTAWU NR 2<sup>1</sup>

### 1. Algorytm sprawdzający, czy sekwencja liczb jest ciągiem graficznym

Dane wejściowe: sekwencja  $n$  liczb naturalnych:  $A = \{A[0], A[1], A[2], \dots, A[n-1]\}$

---

DEGREE\_SEQ( $A, n$ )

```
1. Posortuj tablicę  $A$  nierosnąco
2. while TRUE
3.     do if  $\forall_{i=0,1,\dots,n-1} A[i] == 0$                                 // Jeśli tablica składa się z samych zer,
4.         then return TRUE                                           // to ciąg wejściowy był graficzny.
5.         if  $A[0] < 0$  OR  $A[0] \geq n$  OR  $\exists_{i=1,\dots,n-1} A[i] < 0$ 
6.         then return FALSE
7.         for ( $i = 1$ ;  $i \leq A[0]$ ;  $i = i + 1$ )
8.             do  $A[i] = A[i] - 1$ 
9.          $A[0] = 0$ 
10.        Posortuj tablicę  $A$  nierosnąco
```

---

Dodatkowo: jeśli w wejściowej sekwencji liczba wszystkich nieparzystych elementów jest nieparzysta, to ciąg na pewno nie jest graficzny, ponieważ **liczba wierzchołków o nieparzystym stopniu w grafie prostym jest parzysta.**

### 2. Algorytm oznaczający spójne składowe

---

COMPONENTS( $G$ )

```
1.  $nr = 0$                                                             //  $nr$  - numer spójnej składowej.
2. for każdy wierzchołek  $v$  należący do grafu  $G$ 
3.     do  $comp[v] = -1$                                                 // Wszystkie wierzchołki są nieodwiedzone.
4. for każdy wierzchołek  $v$  należący do grafu  $G$ 
5.     do if  $comp[v] == -1$ 
6.         then  $nr = nr + 1$ 
7.              $comp[v] = nr$  // Oznaczamy  $v$  jako odwiedzony i należący do spójnej składowej  $nr$ .
8.             COMPONENTS_R( $nr, v, G, comp$ ) // Dalsze odwiedzanie: przeszukiwanie w głąb.
9. return  $comp$ 
```

---

Procedura rekurencyjna COMPONENTS\_R( $nr, v, G, comp$ ) ma na celu przeszukanie w głąb zaczynając od danego wierzchołka  $v$ . Wynikiem uruchomienia procedury COMPONENTS( $G$ ) jest tablica  $comp$ ,

---

<sup>1</sup>Na podstawie: Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., *Wprowadzenie do algorytmów*, Wyd. 4, Warszawa, Wydawnictwo Naukowo - Techniczne, 2001, ISBN 83-204-2665-0.

gdzie  $comp[v]$  oznacza numer spójnej składowej, do której należy wierzchołek  $v$ . W trakcie działania algorytmu  $comp[v] == -1$  oznacza, że wierzchołek  $v$  jeszcze nie został odwiedzony.

---

COMPONENTS\_R( $nr, v, G, comp$ )

1. for każdy wierzchołek  $u \in G$  będący sąsiadem  $v$
  2.     do if  $comp[u] == -1$
  3.         then  $comp[u] = nr$
  4.             COMPONENTS\_R( $nr, u, G, comp$ )
- 

### 3. Cykl Eulera<sup>2</sup>

**Cykl Eulera** to zamknięta ścieżka zawierająca każdą krawędź grafu dokładnie jeden raz. Graf spójny jest eulerowski (czyli posiada cykl Eulera) wtedy i tylko wtedy, gdy stopień każdego z jego wierzchołków jest liczbą parzystą. **Mostem** nazywamy taką krawędź, której usunięcie spowoduje utratę spójności.

**Twierdzenie 1. (Poszukiwanie cyklu Eulera: algorytm Fleury’ego)** *Niech  $G$  będzie grafem eulerowskim. Wtedy następująca konstrukcja jest wykonalna i daje w wyniku cykl Eulera w grafie  $G$ .*

*Zaczynij cykl w dowolnym wierzchołku  $u$  i przechodź krawędzie w dowolnej kolejności, dbając jedynie o zachowanie następujących zasad:*

1. *usuwasz z grafu przechodzone krawędzie i wierzchołki izolowane powstające w wyniku usuwania tych krawędzi;*
2. *w każdym momencie przechodź przez most tylko wtedy, gdy nie masz innej możliwości.*

### 4. Cykl Hamiltona

**Cykl Hamiltona** to zamknięta ścieżka zawierająca każdy wierzchołek grafu dokładnie jeden raz (za wyjątkiem pierwszego wierzchołka – on pojawia się dwukrotnie). Warunek konieczny: graf musi być spójny. Warunek konieczny i wystarczający dla dowolnego grafu wejściowego nie istnieje<sup>3</sup> (a przynajmniej nie został jeszcze odkryty). Poszukiwanie cyklu Hamiltona polega na metodzie typu *brute force*: przeszukujemy graf w głąb, odwiedzając wierzchołki dodajemy je na stos. Startujemy od dowolnego wierzchołka.

1. Jeśli stos zawiera wszystkie wierzchołki, to tworzy on ścieżkę Hamiltona: należy jeszcze sprawdzić, czy w grafie jest połączenie między wierzchołkiem startowym a ostatnim na stosie:
  - (a) Jeśli tak, to ścieżka jest cyklem Hamiltona, czyli graf jest hamiltonowski.
  - (b) Jeśli nie, to usuwamy ostatni wierzchołek ze stosu, oznaczamy go jako nieodwiedzony i wycofujemy się na wyższy poziom rekurencji.

---

<sup>2</sup>Źródło twierdzenia 1.: Robin J. Wilson, *Wprowadzenie do teorii grafów*, Wyd. drugie, Warszawa, Wydawnictwo Naukowe PWN, 1998, ISBN 83-01-12641-8.

<sup>3</sup>Dla zainteresowanych – niektóre warunki wystarczające, by graf był hamiltonowski: <https://www.math.uni-hamburg.de/home/diestel/books/graph.theory/preview/Ch10.pdf>. Uwaga: proszę pamiętać, że – skoro są to warunki wystarczające – jeśli graf ich nie spełnia, to nadal nie wiadomo, czy jest hamiltonowski, czy nie.

2. Jeśli stos nie zawiera wszystkich wierzchołków, to rekurencyjnie wywołujemy procedurę dla nieodwiedzonych sąsiadów ostatniego ze stosu. Jeśli po zakończeniu pętli stos nadal nie zawiera wszystkich wierzchołków grafu, to usuwamy ostatni wierzchołek ze stosu, oznaczamy go jako nieodwiedzony i wycofujemy się na wyższy poziom rekurencji.

Jeżeli powyższa metoda siłowa nie zwróci cyklu Hamiltona, to znaczy, że graf nie jest hamiltonowski.