Wstęp do Informatyki 2023/2024
Lista 13

Instytut Informatyki, Uniwersytet Wrocławski

19, 23 i 26 stycznia 2024

Algorytmy rozwiązujące zadania z tej listy mogą mieć postać pseudokodu, w szczególności mogą wykorzystywać abstrakcyjne typy danych jak kolejka, stos, lista wiązana. Można również korzystać z funkcji/algorytmów podanych na wykładzie (sprawdzanie spójności, wyznaczanie składowych spójności, przeglądanie w głąb, przeglądanie wszerz), wskazując ewentualne zmiany, które do nich trzeba wprowadzić. Użycie takich funkcji wymaga jednak podania ich specyfikacji. Zadania dotyczą grafów nieskierowanych, chyba że w treści zadania napisane jest wprost, że zadanie dotyczy grafów skierowanych.
Oszacuj złożoność czasową i pamięciową podanych w rozwiązaniach zadań algorytmów. Uzasadnij ich poprawność.

1. [0.5] Napisz funkcję/algorytm, która przekształca reprezentację grafu w postaci macierzy sąsiedztwa na tablicę list sąsiedztwa, oraz funkcję/algorytm przekształcającą tablicę list sąsiedztwa do postaci macierzy sąsiedztwa.

W poniższych zadaniach przyjmij, że grafy są podawane w postaci list sąsiedztwa.

2. [1] *Mostem* nazywamy krawędź, której usunięcie powoduje zwiększenie liczby składowych spójności grafu. Napisz funkcję/algorytm, która dla podanego na wejściu grafu G oraz pary wierzchołków u, v rozstrzyga, czy krawędź (u, v) jest mostem w G. Twoje rozwiązanie powinno działać w czasie $O(n + m)$, gdzie n to liczba wierzchołków a m to liczba krawędzi grafu.

3. [0.5] Napisz funkcję/algorytm, która dla podanego na wejściu grafu G sprawdza, czy w G występuje (co najmniej jeden) most. Twoje rozwiązanie powinno działać w czasie $O(m(n + m))$, gdzie n to liczba wierzchołków a m to liczba krawędzi grafu.

4. [0.5] Napisz funkcję/algorytm, która dla podanego na wejściu grafu skierowanego G i wierzchołków u, v sprawdza czy istnieje ścieżka łącząca u i v. Algorytm podaje też kolejne wierzchołki na takiej ścieżce (o ile ścieżka istnieje). Twoje rozwiązanie powinno działać w czasie $O(n + m)$, gdzie n to liczba wierzchołków a m to liczba krawędzi grafu.

5. [1] Napisz funkcję/algorytm, która dla podanego na wejściu grafu G i wierzchołków u, v wyznacza długość najkrótszej ścieżki łączącej u i v lub -1, gdy ścieżki łączącej u i v brak. Algorytm podaje też kolejne wierzchołki na takiej ścieżce (o ile ścieżka istnieje).

6. [1] *Punktem artykulacji* nazywamy wierzchołek, którego usunięcie powoduje zwiększenie liczby składowych spójności grafu. Napisz funkcję/algorytm, która dla podanego na wejściu grafu G oraz wierzchołka v rozstrzyga, czy v jest punktem artykulacji grafu G. Twoje rozwiązanie powinno działać w czasie $O(n + m)$, gdzie n to liczba wierzchołków a m to liczba krawędzi grafu.

7. [1] Zbiór wierzchołków V' grafu $G(V, E)$ nazywamy maksymalnym zbiorem niezależnym gdy:

   (a) żadna para wierzchołków u, v ze zbioru V' nie jest połączona krawędzią,

   (b) każdy wierzchołek ze zbioru $V \setminus V'$ jest połączony krawędzią z co najmniej jednym wierzchołkiem ze zbioru V'.

   Podaj algorytm, który znajduje maksymalny zbiór niezależny w czasie $O(n + m)$ oraz sformułuj precyzyjnie specyfikację, którą realizuje Twój algorytm.

8. [1] Napisz funkcję/algorytm, która dla podanego na wejściu grafu G sprawdza, czy w G występuje cykl. W przypadku występowania (co najmniej jednego) cyklu, algorytm zwraca ciąg wierzchołków tworzący cykl w wejściowym grafie.

Zadania dodatkowe, nieobowiązkowe (nie wliczają się do puli punktów do zdobycia na ćwiczeniach, punktacja została podana tylko jako informacja o trudności zadań wg wykładowcy):

1. [2] Napisz funkcję/algorytm, która dla podanego na wejściu grafu skierowanego G sprawdza, czy G jest silnie spójny (p. materiały do wykładu).

2. [2] Napisz funkcję/algorytm, która dla podanego na wejściu grafu skierowanego G wyznacza jego silnie spójne składowe (p. materiały do wykładu).

3. [2] Napisz funkcję/algorytm, która dla podanego na wejściu grafu G wyznacza jego wszystkie punkty artykulacji.

4. [2] Napisz funkcję/algorytm, która dla podanego na wejściu grafu G sprawdza, czy w G występuje (co najmniej jeden) most. Twoje rozwiązanie powinno działać w czasie $O(n + m)$, gdzie n to liczba wierzchołków a m to liczba krawędzi grafu.