

Algorytmy grafowe – reprezentacja, algorytmy przeszukiwania

Zadanie 1

Podaj zalety i wady reprezentacji grafu w postaci:

1. Macierzy sąsiedztwa
2. Listy sąsiedztwa

Zadanie 2

Zdefiniuj struktury danych dla grafu nieskierowanego rzędu 10 (liczba wierzchołków) – opisz

Zadanie 3

- Zaimplementuj wybrany algorytm przeszukiwania grafu (wszerz / w głąb).
- Wierzchołki odwiedzone należy ponumerować: 1 ... n
- Określ jaki warunek musi być spełniony aby wykryć niespójność grafu / acykliczność – warunki te dodaj do kodu wraz z komunikatami odpowiedniej treści
- Wklej kod źródłowy z komentarzami opisującymi realizowane kroki

Zadanie 4

Zdefiniuj dane dla trzech przypadków:

1. Graf spójny acykliczny
2. Gra spójny z cyklami
3. Graf niespójny z cyklami

Zaprezentuj uzyskane wyniki dla powyższych przypadków

Zadanie 5

Podaj propozycje (opis słowny) jak można znaleźć

- Wierzchołek rozpajający grafu
- Centrum grafu
- Jakie inne własności grafu można badać analizując dane i przebieg algorytmu (dodając konieczne zmienne)

Algorytmy

Opisy – np. Wikipedia, <http://www.algorytm.edu.pl/>,
https://eduinf.waw.pl/inf/alg/001_search/0126.php,

BFS (ang. Breadth-first search).

```
Graf G
Wierzchołek s, v
Kolejka FIFO
Numeracja wierzchołków          No[]
Aktualny nr                      No
Własności spójny/niespójny, acykliczny/z cyklami

BFS (G, s)
1. Inicjalizacja zmiennych          // Wyzerowanie No[]
2. Nadanie wierzchołkowi v=s numeru No=1
3. Umieszczenie w FIFO sąsiadów v
4. Dopóki FIFO nie jest pusta
    4a pobranie z FIFO wierzchołka v (z usunięciem)
    4b Nadanie kolejnego numeru ++No
    4c Dodanie nieponumerowanych sąsiadów v do FIFO
5. Analiza No[] - Wyjście: numeracja i własności grafu
```

DFS (ang. Depth-first search).

```
Graf G
Wierzchołek s, u, v
Numeracja wierzchołków          No[]
Aktualny nr                      No
Własności spójny/niespójny, acykliczny/z cyklami

function VisitNode(u):
Nadanie kolejnego numeru ++No
Dla każdego nieponumerowanego sąsiada u wierzchołka s: VisitNode(v)

DFS(G, s):
1. Inicjalizacja zmiennych          // Wyzerowanie No[]
2. Nadanie wierzchołkowi s numeru No=1
3. Dla każdego nieponumerowanego sąsiada u wierzchołka s: VisitNode(u)
```

W DFS zamiast rekurencji może być stosowany stos (bufor LIFO) - operacje:

- push(obiekt) – czyli odłożenie obiektu na stos;
- pop() – ściągnięcie obiektu ze stosu i zwrócenie jego wartości;
- isEmpty() - sprawdzenie czy na stosie znajdują się już jakieś obiekty.

Sprawozdanie

na platformie UPEL (Sprawozdanie 2) w terminie ćwiczeń - pliki edycji doc / pdf, listingi kodu, zrzuty ekranów, pliki wynikowe.

Zadania domowe

Przygotowanie do kolejnych - ćwiczeń 3 – znajomość algorytmu poszukiwania minimalnego drzewa rozpinającego grafu (Dijkstry-Prima)