

Algorytmy i Struktury Danych

Praca samodzielna, Laboratorium 4

Temat: Implementacja wariantu przeszukiwania grafu

Dozwolone języki programowania: C/C++, C#, Java, Python

Wariant zadania: weź ostatnią cyfrę swojego numer indeksu i policz jej resztę z dzielenia przez 4.

0 - Wariant A – BFS, lista sąsiedztwa

1 - Wariant B – DFS, macierz sąsiedztwa

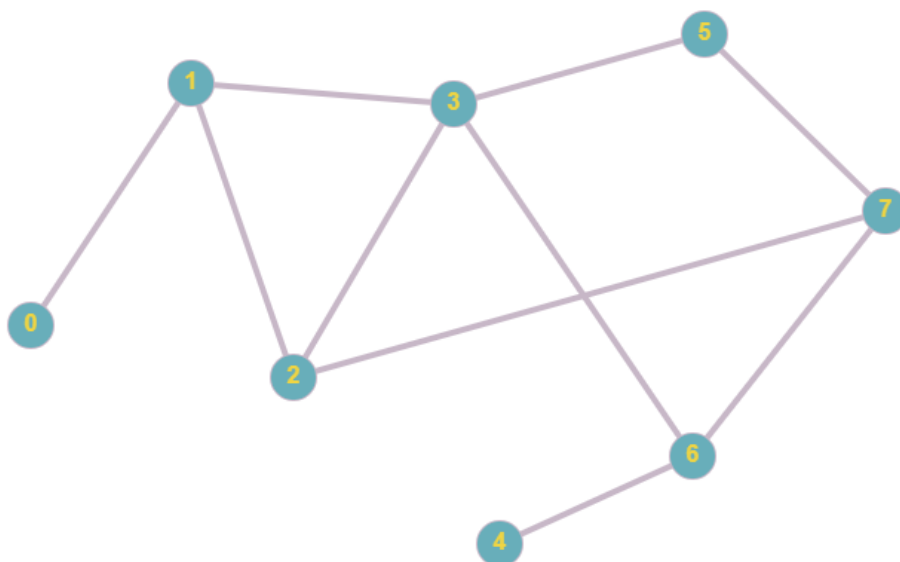
2 - Wariant C – BFS, macierz sąsiedztwa

3 - Wariant D – DFS, lista sąsiedztwa

1. Implementacja swojego wariantu(10 punktów):

- Dla podanego na wejście grafu należy wypisać kolejność BFS/DFS przechowując graf na odpowiedniej strukturze danych.
- Wierzchołkiem początkowym dla algorytmów zawsze jest wierzchołek 0.
- Na początek pobieramy ilość wierzchołków grafu (v) a następnie ilość krawędzi (e). Następnie pobieramy e -razy informację o krawędziach – wierzchołek startowy i końcowy (patrz przykład).
- Przyjmujemy, że graf z wejścia jest nieskierowany i nie ważony. Oznacza to, że krawędź prowadząca z A do B, prowadzi też z B do A.

Przykładowy graf i wejście (8 wierzchołków, 10 krawędzi):



Wejście:

```
8 10
0 1
1 2
1 3
2 3
2 7
3 5
3 6
5 7
4 6
6 7
```

Wyjście BFS:

```
0 1 2 3 7 5 6 4
```

Wyjście DFS:

```
0 1 2 3 5 7 6 4
```

Program nie powinien wyświetlać żadnych innych informacji oprócz zaprezentowanego wyjścia. To bardzo ważne, gdyż program może być testowany automatycznie.

Aby oddać zadanie należy wysłać kod oraz screena z przykładem uruchomienia aplikacji (może być przykładowy graf z instrukcji). Paczkę wysyłamy na platformie moodle.

Wykrycie plagiatu oznacza -10 punktów.

Serwis przydatny do testowania grafów (rysowania, uruchamiania niektórych algorytmów (w tym BFS, DFS), generowania macierzy sąsiedztwa itp):

<https://graphonline.ru/en/>