# Algorytmy Optymalizacji Dyskretnej
# Laboratorium 1

Kacper Błasiak

2025/2026

## Zadanie 1

Celem było zaimplementowanie algorytmów DFS i BFS dla grafów skierowanych i nieskierowanych. Złożoność algorytmów wynosi $O(|V| + |E|)$. Wyniki testów znajdują się w pliku results1.txt.

## Zadanie 2

Zaimplementowano algorytm Kahna służący do sortowania topologicznego grafu skierowanego. Algorytm usuwa kolejno wierzchołki o zerowym stopniu wejściowym, wykrywając przy tym cykle. Złożoność obliczeniowa wynosi $O(|V| + |E|)$. Jako przykład testowy wykorzystano graf acykliczny skierowany o liczbie wierzchołków $|V| = 12$ oraz krawędzi $|E| = 17$, taki, że wszystkie krawędzie prowadzą z wierzchołków o niższym indeksie do wierzchołków o wyższym indeksie. Aby uzyskać graf cykliczny, dodano krawędź $V_7 \to V_1$, co tworzy cykl $V_1 \to V_3 \to V_7 \to V_1$. Wyniki testów znajdują się w pliku results2.txt.

## Zadanie 3

Zrealizowano algorytm Kosaraju do wyznaczania silnie spójnych składowych grafu skierowanego. Algorytm wykorzystuje dwa przebiegi DFS: pierwszy do ustalenia kolejności postorder, a drugi na grafie transponowanym. Złożoność wynosi $O(|V|+|E|)$. Jako pierwszy przykład testowy wykorzystano graf skierowany o liczbie wierzchołków $|V| = 12$ oraz krawędzi $|E| = 17$, taki, że wszystkie krawędzie prowadzą z wierzchołków $V_i$ o niższym indeksie do wierzchołków $V_j$ o wyższym indeksie $(i < j)$.

Drugi przykład przedstawia graf skierowany z trzema wyraźnymi grupami wierzchołków: $C_1 = \{V_1, V_2, V_3, V_4\}$, $C_2 = \{V_5, V_6, V_7, V_8\}$ oraz $C_3 = \{V_9, V_{10}, V_{11}, V_{12}\}$. Każda z tych grup tworzy cykl, w którym z każdego wierzchołka można dotrzeć do każdego innego, czyli stanowi osobną silnie spójną składową. Dodatkowe krawędzie $V_4 \to V_5$ i $V_8 \to V_9$ łączą te części grafu, lecz nie tworzą między nimi pełnej dwukierunkowej łączności, więc cały graf nie jest silnie spójny.

Wyniki testów znajdują się w pliku results3.txt.

# Zadanie 4

Zaimplementowano algorytm sprawdzający dwudzielność grafu przy użyciu BFS i kolorowania wierzchołków. Graf jest dwudzielny, jeśli możliwe jest przypisanie dwóch kolorów w taki sposób, aby żadne dwa sąsiadujące wierzchołki nie miały tego samego koloru. Algorytm działa w czasie $O(|V| + |E|)$ i wypisuje podział na zbiory $V_0$ i $V_1$ dla $n \leq 200$.

Jako przykład grafu dwudzielnego wykorzystano graf nieskierowany o $|V| = 10$ i $|E| = 10$. Zbiór wierzchołków został podzielony na dwie części: $V_0 = \{V_1, V_2, V_3, V_4, V_5\}$ oraz $V_1 = \{V_6, V_7, V_8, V_9, V_{10}\}$, tak że każda krawędź łączy wierzchołki należące do różnych zbiorów. Graf nie zawiera cykli o nieparzystej długości, co potwierdza jego dwudzielność i zgodność z teorią kolorowania grafów.

Jako przykład grafu niedwudzielnego wykorzystano modyfikację powyższego grafu, do którego dodano jedną krawędź $V_6 {-}{-} V_7$. Nowa krawędź łączy dwa wierzchołki z tego samego zbioru $V_1$, co prowadzi do powstania cyklu nieparzystego $V_1 {-}{-} V_6 {-}{-} V_7 {-}{-} V_1$. Obecność takiego cyklu powoduje konflikt kolorów przy kolorowaniu BFS i sprawia, że graf nie spełnia warunku dwudzielności.

Wyniki testów znajdują się w pliku `results4.txt`.