

Sprawozdanie – Projekt 4: Algorytmy z powracaniem

Laboratoria Algorytmów i Struktur Danych

10 czerwca 2025

1 Cel projektu

Celem projektu było zaimplementowanie programu obsługującego grafy nieskierowane w dwóch trybach:

- generowania grafów Hamiltonowskich o zadanym nasyceniu (30% i 70%),
- generowania grafów nie-Hamiltonowskich o nasyceniu 50%.

Na tych grafach wykonywano operacje:

- wypisania grafu,
- wyszukiwania cyklu Eulera,
- wyszukiwania cyklu Hamiltona (algorytm z powracaniem),
- eksportu grafu do formatu LaTeX (TikZ).

2 Reprezentacja grafu

Graf został zaimplementowany jako słownik sąsiedztwa `dict[int, set[int]]`. Reprezentacja ta została wybrana ze względu na:

- szybki dostęp do sąsiadów wierzchołka (czas stały),
- łatwe dodawanie i usuwanie krawędzi,
- prostotę implementacji algorytmów przeszukiwania.

3 Generowanie grafów

3.1 Tryb Hamiltonowski

Tworzony jest losowy cykl Hamiltona obejmujący wszystkie n wierzchołków. Następnie dodawane są krawędzie w losowych trójkątach, aż do osiągnięcia zadanej gęstości. Zapewniono parzysty stopień każdego wierzchołka (co jest warunkiem koniecznym dla istnienia cyklu Eulera).

3.2 Tryb Nie-Hamiltonowski

Tworzony jest graf Hamiltonowski (50% nasycenia), po czym ostatni wierzchołek zostaje odizolowany, eliminując możliwość istnienia cyklu Hamiltona.

4 Operacje na grafie

Program umożliwia użytkownikowi:

1. wypisanie listy sąsiedztwa,
2. sprawdzenie istnienia cyklu Eulera (algorytm Hierholzera),
3. wyszukiwanie cyklu Hamiltona (algorytm z powracaniem z ograniczeniem czasowym),
4. eksport grafu do LaTeX (TikZ).

5 Testy i benchmarki

5.1 Grafy Hamiltonowskie

Wygenerowano grafy o liczbie wierzchołków od 50 do 200 (co 10), dla nasycenia 30%. Zmierzono czas działania algorytmu Eulera i Hamiltona. Wyniki zapisano do `czasy_euler.csv` i `czasy_hamilton.csv`.

5.2 Grafy Nie-Hamiltonowskie

Wygenerowano grafy o $n = 20, 25, 30$. Zmierzono czas działania algorytmu Hamiltona i zapisano do `czasy_niehamilton.csv`.

6 Wykresy

- Wykres 1: czas działania algorytmu cyklu Eulera ($t = f(n)$),
- Wykres 2: czas działania algorytmu Hamiltona dla grafów Hamiltonowskich,
- Wykres 3: czas działania algorytmu Hamiltona dla grafów nie-Hamiltonowskich.

7 Obserwacje

- Algorytm Eulera działa bardzo szybko i jego złożoność czasowa jest liniowa względem liczby krawędzi.
- Algorytm Hamiltona działa znacznie wolniej, dla dużych grafów czas działania znacząco rośnie.
- Dla grafów nie-Hamiltonowskich algorytm nie znajduje cyklu i zużywa pełny limit czasu.

8 Eksport grafów do LaTeX

Program umożliwia eksport wygenerowanego grafu do pliku `.tex` z użyciem środowiska `tikzpicture`. Grafy są rozmieszczane na okręgu o stałym promieniu dla zachowania przejrzystości.

9 Staranność wykonania

Kod spełnia zalecenia:

- czytelne nazwy zmiennych,
- brak duplikacji kodu,
- modularność (`main.py`, `graph.py`, `export.py`, `benchmark.py`),
- komentarze w kluczowych miejscach.

10 Wnioski

Projekt realizuje operacje na grafach z uwzględnieniem algorytmów klasy NP-trudnych (cykl Hamiltona) i liniowych (cykl Eulera). Efektywność algorytmów zależy od struktury grafu oraz nasycenia. Algorytmy z powracaniem są niepraktyczne dla dużych grafów bez heurystyk ograniczających przestrzeń przeszukiwań.