

Dokumentacja funkcjonalna

Projekt w języku C: Podział grafu na części

Autor: Mateusz Michalski, Paweł Kozłowski

22 Kwiecień, 2025

Cel projektu

Głównym celem projektu jest stworzenie praktycznego narzędzia, które umożliwia użytkownikowi podział grafu na zadana liczbę części w sposób możliwie najbardziej optymalny. W tym kontekście optymalność oznacza zachowanie możliwie równego rozkładu wierzchołków pomiędzy poszczególne części oraz minimalizację liczby krawędzi, które muszą zostać przecięte w procesie dzielenia grafu.

Zastosowanie takiego programu może mieć miejsce w różnych dziedzinach informatyki, takich jak analiza sieci, przetwarzanie danych, projektowanie systemów równoległych, optymalizacja zasobów czy zarządzanie infrastrukturą techniczną. Dodatkowo projekt ten rozwija umiejętności związane z implementacją algorytmów grafowych oraz z obsługą pamięci dynamicznej w języku C.

Funkcje programu

Program posiada zestaw funkcjonalności, które umożliwiają jego uniwersalne wykorzystanie:

- wczytywanie grafu z pliku wejściowego w specjalnym formacie opartym na reprezentacji CSR (Compressed Sparse Row) z możliwością definiowania grup wierzchołków,
- analizowanie struktury grafu pod kątem spójności i potencjalnych możliwości podziału,
- iteracyjne dzielenie grafu na części przy zachowaniu ograniczeń dotyczących marginesu rozmiarów oraz minimalizacji liczby przecięć krawędzi,
- zapis wyników podziału do pliku tekstowego lub binarnego w zależności od preferencji użytkownika,
- elastyczne sterowanie działaniem programu za pomocą parametrów przekazywanych w linii komend.

Argumenty wywołania

Program przyjmuje poniższe parametry uruchomieniowe:

- n <liczba> określa liczbę części, na które graf ma zostać podzielony. Jeżeli nie zostanie podany, domyślnie wartość ta wynosi 2.
- m <liczba> definiuje maksymalny margines procentowy różnicy pomiędzy rozmiarami części. Domyślnie wynosi 10%.
- o <plik> umożliwia zapis wyników działania programu do wskazanego pliku tekstowego lub binarnego.
 - t nakazuje wypisanie wyników jedynie na terminalu, bez zapisu do pliku.
 - b wymusza zapis do pliku w formacie binarnym.

plik wejściowy ostatni parametr wskazuje ścieżkę do pliku zawierającego dane wejściowe opisujące graf.

Format pliku wejściowego

Dane wejściowe muszą być zapisane w formacie tekstowym zgodnym z wymaganą strukturą. Plik powinien zawierać:

1. liczbę maksymalnych sąsiadów w pojedynczym wierszu (całkowita liczba),
2. listę sąsiadów dla wszystkich wierzchołków, zapisana jako ciąg wartości oddzielonych średnikami (np. 0;2;4;5),
3. listę wskaźników CSR, również oddzielona średnikami (np. 0;3;5;8), która wskazuje, od którego indeksu w tablicy sąsiadów zaczynają się dane dla danego wierzchołka,
4. (opcjonalnie) listę grup, definiującą zbiorcze powiązania między wierzchołkami,
5. (opcjonalnie) wskaźniki grup.

Graf może być zarówno zwykłym grafem nieskierowanym, jak i zawierać dodatkowe informacje o przynależności do grup.

Obsługa błędów

Program przewiduje różne sytuacje wyjątkowe i reaguje na nie odpowiednimi komunikatami:

- brak podania pliku wejściowego skutkuje komunikatem: **Nie podano pliku wejściowego!** i zakończeniem działania programu (kod wyjścia: 1),
- problemy z otwarciem pliku powodują wypisanie błędu: **Błąd przy otwieraniu pliku wejściowego**,
- w przypadku nieudanej alokacji pamięci dynamicznej program wypisuje specyficzny komunikat zależny od miejsca wystąpienia problemu i kończy działanie,
- gdy nie jest możliwe wykonanie podziału spełniającego warunki marginesu i spójności, program informuje o tym użytkownika i kontynuuje analizę.

Przykład uruchomienia

Przykładowe wywołanie programu może wyglądać następująco:

```
./program -n 3 -m 15 -o wynik.txt graf.txt
```

Oznacza to podział grafu wczytanego z pliku `graf.txt` na 3 części, z dopuszczalnym marginesem 15%, a wynik zostanie zapisany do pliku tekstowego `wynik.txt`.