

Dokumentacja implementacyjna

Projekt w języku C: Podział grafu na części

Autor: Mateusz Michalski, Paweł Kozłowski

22 Kwiecień, 2025

Struktura projektu

Program został zorganizowany w sposób modułowy, co pozwala na łatwą nawigację w kodzie, testowanie poszczególnych komponentów oraz ewentualną rozbudowę w przyszłości. Zastosowanie podziału na pliki `.c` i `.h` zwiększa przejrzystość kodu źródłowego i ułatwia jego konserwację. Struktura plików przedstawia się następująco:

- `main.c` – plik główny programu. Zawiera logikę związaną z obsługą parametrów wejściowych, otwieraniem plików oraz inicjalizacją głównych struktur danych. To właśnie tutaj rozpoczyna się wykonanie programu.
- `dzielenie.c/h` – moduł zawierający algorytmy odpowiedzialne za analizę spójności podgrafów, ich podział z uwzględnieniem marginesu oraz rejestrację przeciętych krawędzi. Plik nagłówkowy zawiera deklaracje struktur i funkcji eksportowanych.
- `pliki.c/h` – moduł przetwarzający dane wejściowe. Wczytuje dane z pliku tekstowego w formacie CSR oraz konwertuje je do dynamicznej macierzy sąsiedztwa.
- `test.c` – zestaw testów jednostkowych weryfikujących poprawność działania funkcji pomocniczych oraz algorytmów.

Specyfikacja wejścia

Dane wejściowe wczytywane są z pliku tekstowego. Plik ten musi być zgodny z ustalonym formatem i zawierać:

1. liczbę maksymalnych sąsiadów w wierszu – wykorzystywana do optymalizacji alokacji pamięci,
2. listę sąsiadów – zawierająca kolejno wszystkie sąsiedztwa wierzchołków w kolejności CSR, oddzielone średnikami,
3. listę wskaźników wierszy – definiująca początkowe indeksy sąsiadów dla każdego wierzchołka,
4. (opcjonalnie) listę grup – wykorzystywana do grupowania wierzchołków w celu automatycznego tworzenia krawędzi między nimi,
5. (opcjonalnie) wskaźniki do grup.

Plik może zawierać również puste linie lub komentarze, które program pomija podczas przetwarzania.

Specyfikacja wyjścia

Po przeprowadzeniu operacji dzielenia grafu, program generuje dane wyjściowe w jednym z dwóch trybów:

- jako plik tekstowy zawierający dane czytelne dla człowieka,
- jako plik binarny, jeśli użytkownik wybrał flagę `-b`.

Zawartość pliku wynikowego obejmuje:

- liczbę początkowych składowych spójnych w grafie,
- kolejne wykonane podziały wraz z rozmiarami podgrafów i marginesami,
- przecięte krawędzie – wypisane w formacie `u -- v`,
- końcowa struktura podgrafów – w tym listę wierzchołków przypisaną do każdego z nich.

Format pliku tekstowego pozwala również na jego ponowne przetworzenie w kolejnym uruchomieniu programu.

Struktury danych

W celu zwiększenia przejrzystości oraz umożliwienia wielokrotnego wykorzystania kodu, zastosowano poniższe struktury:

- **Podgraf** – zawiera dynamiczną tablicę wierzchołków oraz rozmiar podgrafu. Umożliwia operacje takie jak iteracyjne dzielenie czy sprawdzanie spójności.
- **Krawedz** – struktura pomocnicza do przechowywania par wierzchołków, których połączenia zostały przecięte. Przydatna przy analizie wpływu podziału na strukturę grafu.
- **macierz sąsiedztwa** – dynamicznie alokowana struktura `int**`, reprezentująca graf w postaci nieskierowanej.

Dodatkowo stosowano pomocnicze zmienne logiczne i buforowe (np. `odwiedzone`, `kolejka`) wewnątrz algorytmów DFS i BFS.

Szczegóły techniczne

Program został napisany w języku C z zachowaniem zasad bezpieczeństwa i czytelności. Poniżej kluczowe aspekty techniczne:

- Parametry wejściowe analizowane są za pomocą biblioteki `getopt()`, co pozwala na elastyczną obsługę flag i wartości domyślnych.
- Alokacja pamięci odbywa się za pomocą `calloc()` i `malloc()` z kontrolą poprawności (sprawdzane wskaźniki `null`).
- Każda funkcja została zaprojektowana jako możliwie niezależna i odpowiedzialna za jedno zadanie (tzw. zasada jednej odpowiedzialności).
- Kod obsługuje błędy zarówno krytyczne (np. brak pamięci), jak i logiczne (np. brak możliwości dalszego podziału podgrafu).
- Wszystkie operacje na plikach są zakończone poprawnym `fclose()` i `free()` dla alokowanych danych.

Możliwości rozbudowy

Projekt posiada potencjał rozbudowy o:

- algorytmy podziału kierowane heurystykami lub algorytmy losowe,
- interfejs graficzny lub wizualizacje wyników za pomocą zewnętrznych bibliotek,
- dodatkowe metryki jakości podziału (np. modularność, centralność).