Dokumentacja funkcjonalna

Dzielenie grafów

Michał Kozak, Jakub Lipiński

10.03.2025

# Cel Projektu:

Celem projektu jest opracowanie oprogramowania w języku C umożliwiającego efektywny podział grafu na określoną liczbę części zgodnie z zadanymi kryteriami. Aplikacja będzie działać jako narzędzie uruchamiane z linii poleceń, przyjmując wszystkie parametry wejściowe jako argumenty programu.

Podział grafu ma minimalizować liczbę przeciętych krawędzi oraz zapewniać względną równomierność rozkładu wierzchołków w każdej części. Dane wejściowe i wyniki działania programu będą zapisywane w pliku tekstowym lub binarnym, umożliwiając ich ponowne wykorzystanie w kolejnych operacjach.

# Opis funkcji udostępnianych przez program:

Program do podziału grafu na części będzie oferował następujące funkcjonalności:

1. **Wczytywanie grafu z pliku**
   * Program przyjmuje graf wejściowy zapisany w formacie tekstowym.
   * Plik wejściowy zawiera listę wierzchołków i krawędzi opisujących strukturę grafu.
   * Obsługa błędnych lub niekompletnych danych wejściowych.
2. **Podział grafu na określoną liczbę części**
   * Użytkownik określa liczbę części, na które graf ma zostać podzielony.
   * Domyślnie graf dzielony jest na 2 części, jeśli użytkownik nie poda innej wartości.
   * Program zapewnia, że liczba wierzchołków w poszczególnych częściach nie różni się o więcej niż zadany margines procentowy (domyślnie 10%).
   * Podział minimalizuje liczbę przeciętych krawędzi.
3. **Obsługa parametrów z linii poleceń**
   * Wszystkie opcje programu są konfigurowane poprzez argumenty przekazywane w linii poleceń.
   * Użytkownik może określić m.in. ścieżkę do pliku wejściowego, liczbę części oraz dopuszczalny margines różnicy w liczbie wierzchołków.
4. **Zapisywanie wyników do pliku**
   * Program umożliwia zapis wyników działania w formacie tekstowym lub binarnym.

# Dane wejściowe:

Dane wejściowe to plik, który zawiera 5 linii. Pierwsze trzy linie opisują położenie poszczególnych wierzchołków w przestrzeni dwuwymiarowej jako postać macierzowa. Dwie ostatnie linijki opisują zależności pomiędzy danymi wierzchołkami.

Szczegółowe znaczenia każdej linijki:

1. Maksymalna wielkość wiersza w macierzy. Czyli maksymalna liczba kolumn, która nie zawsze jest równa liczbie maksymalnej a jedynie mniejsza bądź równa.
2. Każda kolejna liczba oznacza kolejny węzeł (wierzchołek), a wartość liczby oznacza indeks wiersza, w którym dany węzeł będzie się znajdować (czyli w której kolumnie, licząc od 0). Ilość wszystkich liczb w tej linii oznacza ilość węzłów (wierzchołków).
3. Każde po kolei dwie liczby oznaczają zakres w którym znajdują się indeksy (czyli numerki kolumn) dla kolejnych wierszy macierzy. Ilość – 1 oznacza liczbę ogólna wierszy w macierzy.
4. Węzły są zapisane w grupach, gdzie każda grupa reprezentuje zbiór węzłów należących do jednego fragmentu grafu.
5. (opcjonalnie 6,7,… (kolejna)) Oznaczają wskaźniki na pierwszą pozycje węzła (wierzchołka) dla każdej grupy w liście krawędzi z punktu 4.

# Argumenty wywołania programu:

|  |  |  |  |
| --- | --- | --- | --- |
| Flaga | Argument | Opis | Wartości domyślne i dopuszczalne |
| -i | [plik\_wejściowy] | |  | | --- | | Ścieżka do pliku tekstowego zawierającego definicję grafu. |  |  | | --- | |  | | |  | | --- | | Wymagane. Musi być poprawną ścieżką do pliku. |  |  | | --- | |  | |
| -o | [plik\_wyjściowy] | |  | | --- | | Ścieżka do pliku, w którym zapisany zostanie wynik działania programu. |  |  | | --- | |  | | |  | | --- | | Wymagane. Musi być poprawną ścieżką do pliku. |  |  | | --- | |  | |
| -p | [liczba\_części] | |  | | --- | | Liczba części, na które ma zostać podzielony graf. |  |  | | --- | |  | | |  | | --- | | Wymagane. Wartość całkowita ≥ 2. Domyślnie 2. |  |  | | --- | |  | |
| -m | [margines]] | |  | | --- | | Maksymalna dopuszczalna różnica procentowa w liczbie wierzchołków między częściami. |  |  | | --- | |  | | |  | | --- | | Opcjonalne. Wartość liczbową podaje się w procentach (np. 10 oznacza 10%). Domyślnie 10%. |  |  | | --- | |  | |
| -h | (null) | |  | | --- | | Wyświetla pomoc i kończy działanie programu. |  |  | | --- | |  | | Brak wartości, program wypisuje opis dostępnych flag. |

# Dane wyjściowe:

Wynikiem działania programu będzie podział grafu na zadaną liczbę części zgodnie z określonymi kryteriami. Program zwróci informacje o podziale, w tym przypisanie wierzchołków do poszczególnych części oraz liczbę przeciętych krawędzi. Dodatkowo uwzględni statystyki dotyczące równomierności podziału, czyli procentową różnicę w liczbie wierzchołków między największą a najmniejszą częścią.

Przykład:

Polecenie: ./program -i graph.txt -o result.txt -p 2 -m 10

Plik wejściowy: (graph.txt)

4

0;1;3;2;3;0;1;1;2;3

0;3;5;7

0;1;5;3;2;5;8;6;4;6;7;5;8;6;9;7;9

0;4;8;11;13;15

Plik wyjściowy: (result.txt):

4

0;1;3;2;3;0;1;1;2;3

0;3;5;7

[PUSTY WIERSZ]

0;1;5;3;5;8

0;4

[PUSTY WIERSZ]

2;6;6;4;9;4;7

0;2;5

### Krótki opis pliku wyjściowego:

Pierwsze trzy linijki bez zmian, tak samo jak w wejściowym. Kolejne linijki po pustym wierszu oznaczają podzielone grafy oraz zależności pomiędzy nimi dokładanie takie samo jak 4 i 5 linijka z pliku wejściowego.

# Teoria:

**Algorytm BFS**Algorytm BFS (Breadth-First Search) to metoda przeszukiwania grafu, która eksploruje go poziomami – najpierw odwiedzając wszystkie wierzchołki na danym poziomie, zanim przejdzie do kolejnego. BFS działa na zasadzie kolejki: zaczynamy od wybranego węzła startowego, oznaczamy go jako odwiedzony i dodajemy do kolejki. Następnie pobieramy pierwszy element z kolejki, sprawdzamy jego sąsiadów i jeśli nie byli jeszcze odwiedzeni, dodajemy ich do kolejki i oznaczamy jako odwiedzonych. Proces ten powtarzamy, aż kolejka stanie się pusta, co oznacza, że odwiedziliśmy wszystkie możliwe wierzchołki dostępne z punktu startowego.

Algorytm ten ma złożoność O(V + E), gdzie V to liczba wierzchołków, a E liczba krawędzi. Działa efektywnie na grafach rzadkich i pozwala m.in. na znalezienie najkrótszej ścieżki w grafie nieskierowanym, sprawdzanie spójności grafu, a także podział grafu na mniejsze regiony. W kontekście naszego projektu BFS może być używany do analizy struktury połączeń oraz podziału dużego grafu na mniejsze, łatwiejsze do przetwarzania fragmenty, co pozwala na optymalizację obliczeń i lepszą organizację danych.

# Komunikaty Błędów:

|  |  |  |
| --- | --- | --- |
| **Numer błędu** | **Opis sytuacji** | **Komunikat błędu** |
| **1** | Błędna liczba argumentów w linii poleceń | "Błąd: Nieprawidłowa liczba argumentów. Proszę podać wszystkie wymagane argumenty." |
| **2** | Zła wartość liczby części (k) | "Błąd: Liczba części (k) musi być liczbą całkowitą większą od 0 i mniejszą lub równą liczbie wierzchołków." |
| **3** | Zły margines procentowy | "Błąd: Margines procentowy musi być liczbą całkowitą w przedziale od 0 do 100." |
| **4** | Błędny format pliku wejściowego | "Błąd: Plik wejściowy nie jest w poprawnym formacie. Upewnij się, że plik zawiera poprawny opis grafu." |
| **5** | Plik wejściowy nie istnieje | "Błąd: Nie znaleziono pliku wejściowego. Proszę sprawdzić ścieżkę pliku." |
| **6** | Błędna wartość w pliku wejściowym (np. brak wierzchołków, błędne krawędzie) | "Błąd: Plik wejściowy zawiera błędne dane. Sprawdź poprawność formatu grafu." |
| **7** | Nieprawidłowa opcja flagi w linii poleceń | "Błąd: Nieznana flaga lub opcja w linii poleceń. Proszę sprawdzić dostępne opcje." |
| **8** | Błąd przy próbie zapisu do pliku wynikowego | "Błąd: Nie udało się zapisać pliku wynikowego. Sprawdź uprawnienia do zapisu." |
| **9** | Niedostateczna liczba wierzchołków do podziału | "Błąd: Graf ma zbyt mało wierzchołków, aby wykonać żądany podział na części." |