Dokumentacja Funkcjonalna
Podział grafu

Anastasiia Dmytrenko, Arkadiusz Perko

13.03.2025

1 Cel projektu

Celem projektu jest implementacja narzędzia matematycznego w języku C, które automatyzuje i upraszcza proces rozwiązywania problemu podziału grafu. Aplikacja ma na celu efektywny podział wierzchołków grafu na równomiernie wielkościowe części przy minimalnej liczbie krawędzi między tymi grupami, co jest kluczowe dla zachowania strukturalnej spójności grafu.

2 Dane wejściowe

Program oczekuje danych wejściowych w formacie tekstowym, zapisanych w pliku opisującym graf za pomocą listy sąsiedztwa. Dane powinny być przygotowane według precyzyjnie zdefiniowanego schematu:

- **Pierwsza linia:** Maksymalna możliwa liczba węzłów w wierszu (w grafie nie musi znajdować się wiersz o takiej liczbie węzłów).
- **Druga linia:** Indeksy węzłów w poszczególnych wierszach - liczba wszystkich indeksów odpowiada liczbie węzłów grafu.
- **Trzecia linia:** Wskaźniki na pierwsze indeksy węzłów w liście wierszy z punktu 2.
- **Czwarta linia:** Lista sąsiedztwa węzłów grafu.
- **Piąta linia (i każda kolejna w przypadku grafów niespójnych):** Wskaźniki na pierwsze węzły w poszczególnych grupach z linii 4.

Przykładowa zawartość pliku wejściowego:

```
4
0; 2; 3; 1; 1; 2
0; 0; 3; 4; 6
0; 1; 1; 2; 3; 2; 3; 5; 3; 4
0; 2; 5; 8; 10
```

3 Argumenty wywołania programu

Program obsługuje serię argumentów wywołania, które pozwalają użytkownikowi na szczegółową konfigurację procesu.

3.1 Standardowe wywołanie

```
./graph_partition -i [input-file] -o [output-file] -r [format-output-file] -p [parts] -b [error-margin] -m [method(optional)] [other]
```

- **input-file** - Ścieżka do pliku z danymi grafu.
- **output-file** - Ścieżka, gdzie zostanie zapisany plik z wynikiem działania programu.
- **format-output-file** - Określenie jaki format ma mieć plik wyjściowy (ASCII lub binarny).
- **parts** - Liczba części na które graf ma zostać podzielony.
- **error-margin** - Dopuszczalny margines błędu dla równomierności rozmiarów części (przyjmujemy wartość w punktach procentowych).
- **method** - (opcjonalnie) Metoda podziału grafu, może być niepodana, wtedy użyta zostanie metoda domyślna (*Algorytm Kernighan-Lin).* Do wyboru są 3 metody: *-kl, -m, -w.*
- **other** - Pozostałe opcjonalne flagi (*-f, –force* lub *-h, –help*).

3.2 Dostępne flagi

-f, –force - Zmusza program do podziału grafu nawet w przypadku braku możliwości zastosowania podanego marginesu błędu do rozwiązania.

-h, –help - Wyświetla instrukcję użytkowania programu.

-m, –method - Wybiera algorytm spośród wartości:

kl - Wybiera metodę opartą na algorytmie Kernighan-Lin do podziału grafu.

m - Wybiera metodę opartą na macierzy Laplaciana do podziału grafu.

w - Wybiera metodę opartą na obliczeniu współczynnika wydajności podziału grafu.

-i, –input-file - Wczytuje plik wejściowy znajdujący się pod podaną ścieżką.

-o, –output-file - Zapisuje wynik do pliku w folderze o podanej ścieżce.

-r, –format ascii / binary - Określa format pliku wyjściowego: ASCII lub binarny.

-b - Ustawia dopuszczalny margines błędu dla równomierności rozmiarów części grafu.

-p, –parts - Określa, na ile części ma zostać podzielony graf.

3.3 Przykładowe wywołania

```
./graph_partition -i data.txt -o partition.txt -r ascii -p 4 -b 5 -m kl
```

To wywołanie uruchomi program, który wczyta graf z pliku *data.txt*, podzieli go na *4* części, z marginesem błędu równomierności na poziomie *5%*, używając algorytmu *Kernighan-Lin* do podziału. Wynik zostanie zwrócony w pliku partition.txt w formacie tekstowym.

3.4 Zaawansowane użycie:

Można łączyć różne flagi, aby dostosować działanie programu do szczególnych potrzeb:

```
./graph_partition --input-file data.txt -o partition.txt -r binary -p 3
    -b 10 -m w --force
```

To wywołanie uruchomi podział grafu z użyciem algorytmu opartego na *współczynniku wydajności*, zmuszając program do podziału pomimo potencjalnych trudności w spełnieniu marginesu błędu w *10%*. Wynik zostanie zwrócony w pliku partition.txt w formacie binarnym.

4 Opis funkcji

1. Wczytywanie danych grafu

Służy do wczytywania danych grafu z pliku tekstowego w formacie listy sąsiedztwa.

Dane wejściowe: plik tekstowy z listą sąsiedztwa.

Wynik działania: Graf reprezentowany w pamięci programu, gotowy do przetworzenia.

2. Podział grafu

Funkcja dzieli graf na określoną przez użytkownika liczbę części, starając się równomiernie rozdzielić wierzchołki i minimalizować liczbę przecinających się krawędzi.

Program oferuje trzy metody podziału grafu, z których użytkownik może wybrać przez argumenty wywołania:

2.1. Algorytm Kernighan-Lin

Algorytm heurystyczny optymalizujący podział grafu poprzez minimalizację liczby krawędzi między podzielonymi grupami.

2.2. Metoda macierzy Laplaciana

Wykorzystuje macierz Laplaciana grafu do optymalizacji podziału, bazując na analizie wartości własnych i wektorów własnych tej macierzy.

2.3. Algorytm własny z wykorzystaniem współczynnika wydajności podziału

Implementuje własny algorytm oparty na obliczeniach wydajności podziału, który może być dostosowany do potrzeb użytkownika.

3. Obsługa argumentów wywołania

Funkcja interpretuje i przetwarza argumenty podane przez użytkownika w linii poleceń.

Dane wejściowe: Argumenty wywołania programu, takie jak liczba części na które podzielić graf, nazwa pliku danych wejściowych, nazwa pliku wyjściowego itp.

Wynik działania: Konfiguracja działania programu zgodna z podanymi parametrami.

4. Zapis wyniku do pliku

Funkcja zapisuje wynik podziału grafu do pliku, który może być użyty do ponownego rozwiązywania lub wizualizacji w JAVA.

Dane wejściowe: Wyniki działania funkcji podziału grafu.

Wynik działania: Plik (tekstowy lub binarny) zawierający wyniki podziału grafu.

5 Obsługa błędów

Program monitoruje różne potencjalne błędy podczas działania. Poniżej przedstawiono scenariusze wyjątkowe wraz z odpowiadającymi im komunikatami błędów i zalecanymi działaniami.

1. **Liczba części na które użytkownik chce podzielić graf jest mniejsza niż 2.**
   Komunikat : *"Błąd: liczba części musi być większa lub równa 2."*
   Działanie: Program prosi o ponowne wpisanie liczby części na wartość >=2.
   Kod błędu: 10
2. **Brakujący parametr wywołania**
   Komunikat: *"Błąd: parametry wywołania są niewystarczające, aby uruchomić program."*
   Działanie: Program przerywa działanie. Użytkownik musi ponownie uruchomić program z argumentami wywołania zgodnie z dokumentacją.
   Kod błędu: 11
3. **Nieistniejący parametr**
   Komunikat: *"Błąd: nieznany parametr."*
   Działanie: Program przerywa działanie. Użytkownik musi sprawdzić dostępne parametry w dokumentacji i je dostosować w ponownym uruchomieniu programu.
   Kod błędu: 12
4. **Niepoprawny format pliku wejściowego**
   Komunikat: *"Błąd: niepoprawny format pliku wejściowego."*
   Działanie: Program przerywa działanie. Użytkownik musi upewnić się, że plik wejściowy jest sformatowany zgodnie z wymaganiami programu.
   Kod błędu: 13
5. **Błędne wprowadzanie danych**
   Komunikat: *"Błąd: błędne dane wejściowe."*
   Działanie: Program przerywa działanie. Użytkownik musi zweryfikować i skorygować dane wejściowe.
   Kod błędu: 14
6. **Błąd pamięci**
   Komunikat: *„Błąd: pomyłka pamięci."*
   Działanie: Program przerywa działanie z powodów błędu pamięci.
   Kod błędu: 15