

Projekt JIMP

Zuzanna Pyzikiewicz, Anna Guz

March 2025

1 Cel projektu i opis działania programu

Celem projektu jest stworzenie aplikacji do podziału grafu na określoną liczbę części, przy jednoczesnej minimalizacji liczby przeciętych krawędzi. Liczba wierzchołków otrzymanych części może się różnić o zadany przez użytkownika próg. Aplikacja jest napisana w języku C i działa w trybie nieinteraktywnym. Użytkownik dostarcza plik z grafem i parametry dotyczące podziału. Na tym etapie istnieje również możliwość wyboru formatu danych wyjściowych. Jeśli określona przez użytkownika operacja jest możliwa, program zwraca plik zawierający grafy powstałe po podziale. W innym przypadku pojawia się stosowny komunikat informujący o błędzie.

2 Opis funkcji

Program dzieli graf na zadaną przez użytkownika liczbę części, starając się zminimalizować liczbę przeciętych krawędzi. W tym celu pobiera on sformatowany plik z opisem grafu, a następnie, korzystając z algorytmu, dzieli go na wybraną ilość części. Algorytm uwzględnia również zbalansowanie podziału, dążąc do równomiernego rozłożenia wierzchołków między częściami, a po przeanalizowaniu grafu, gwarantuje znalezienie i zwrócenie optymalnego rozwiązania, minimalizującego liczbę przeciętych krawędzi. Program generuje wyniki w dwóch formatach: tekstowym lub binarnym, w zależności od wyboru użytkownika za pomocą flagi. W przypadku wyboru formatu binarnego, program wewnętrznie konwertuje dane do tego formatu.

3 Dane wejściowe i dostępne flagi

Program działa w trybie nieinteraktywnym, dlatego wszystkie potrzebne informacje otrzymuje w argumentach wejścia. Przy uruchamianiu programu podajemy niezbędne argumenty: liczbę części, na które graf podany w pliku wejściowym ma zostać podzielony; margines procentowy, opisujący maksymalną różnicę między liczbami wierzchołków we wszystkich częściach podzielonego grafu; nazwę pliku wejściowego (w formacie tekstowym) oraz dostępne flagi, przy pomocy których wybieramy format pliku wyjściowego:

- t - zwracany plik wyjściowy będzie w formacie tekstowym,
- b - plik wyjściowy zwracany jest w formacie binarnym.

Format linii wywołania programu:

```
./podziel_graf nazwa_pliku_wejscowego ilosc_czesci margines_procentowy  
[flaga]
```

W przypadku niepodania argumentów wejściowych program wykonuje podział dla:

- ilość_części = 2
- margines_procentowy = 10
- flaga = -t

Program nie działa bez podania pliku wejściowego.

Przykładowe parametry wywołania programu:

```
./podziel_graf plik.txt 2 20 -t
```

- podziel_graf - nazwa programu uruchamianego
- 2 - liczba części, na które zostanie podzielony graf
- 20 - margines procentowy
- -t - flaga, sygnalizująca wybór formatu pliku wyjściowego jako tekstowy

Format pliku wejściowego

Plik wejściowy powinien zawierać 5 informacji, potrzebnych do działania programu, każdą oddzieloną znakiem nowej linii:

1. Maksymalna ilość wierzchołków w wierszu.
2. Macierzowy opis każdego z wierszy grafu.
3. Wskaźniki na początek wiersza z drugiej linii pliku.
4. Połączone grupy węzłów (wypisujemy główny wierzchołek oraz wszystkie wierzchołki, z którymi jest on połączony, po czym przechodzimy do następnego wierzchołka z kolei i powtarzamy, dopóki nie zapiszemy wszystkich relacji).
5. Wskaźniki na wierzchołki główne z czwartej linii pliku.

W przypadku wielu grafów w pliku powyższy punkt piąty składa się z tylu linii, ile jest grafów. Każda nowa linia opisuje krawędzie między wierzchołkami kolejnego grafu.

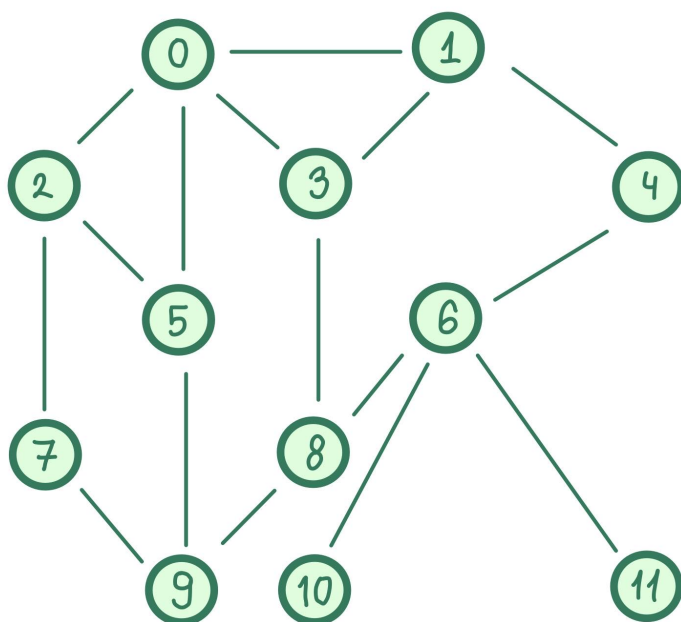
Przykładowy plik wejściowy

```
5
1;3;0;2;4;1;3;0;2;1;2;4
0;2;5;7;9
0;1;2;3;1;3;4;2;5;7;3;8;4;6;5;9;6;8;10;11;7;9;8;9
0;4;7;10;12;14;16;20;22
```

Pierwsze trzy linijki pliku opisują macierz grafu. Dla powyższego przykładu wygląda ona następująco:

$$\begin{bmatrix} 0. & 1. & 0. & 1. & 0. \\ 1. & 0. & 1. & 0. & 1. \\ 0. & 1. & 0. & 1. & 0. \\ 1. & 0. & 1. & 0. & 0. \\ 0. & 1. & 1. & 0. & 1. \end{bmatrix}$$

Podana macierz opisuje poniższy graf



Kolejne linijki pliku opisują krawędzie między wierzchołkami. Na przykładzie mamy opisany pierwszy wierzchołek główny - 0, oraz podane wszystkie wierzchołki sąsiadujące z nim, czyli 1, 2 oraz 3. Kolejny wierzchołek główny jest to 1, co możemy zauważyć po przebadaniu ostatniej linii pliku. Indeks drugiego wierzchołka głównego to cztery, licząc od 0, więc będzie to właśnie 1. Na podstawie czwartej linii pliku możemy odczytać sąsiadów tego wierzchołka - 3 oraz 4. Mimo że sąsiadem 1 jest także 0, nie wypisujemy tego wielokrotnie. Było to już zapisane przy podawaniu sąsiadów wierzchołka głównego 0.

4 Dane wyjściowe

Program zwraca dane w postaci pliku tekstowego lub binarnego, w zależności od preferencji użytkownika. Plik wyjściowy przypomina plik wejściowy, ale w przypadku udanego podziału grafu składa się z więcej niż 5 linii.

Plik wyjściowy zawiera 3 pierwsze linie z pliku wejściowego. Czwarta linia opisuje grupy węzłów połączone za pomocą krawędzi. Linie od 5 do ostatniej zawierają wskaźniki na pierwsze węzły w poszczególnych grupach zawartych w linii 4. Każda z linii o numerach od 5 w górę odnosi się do jednego grafu z tych, na które podzielono oryginalny graf. Plik wyjściowy nie zawiera danych o połączeniach węzłów w oryginalnym grafie.

Jeśli oryginalny plik zawierał kilka grafów i udało się podzielić więcej niż jeden z nich, plik wyjściowy wygląda analogicznie jak przy podziale jednego grafu - linijki od 5 do ostatniej opisują grafy powstałe po podziale kolejnych grafów.

5 Sytuacje Wyjątkowe

Wśród możliwych sytuacji wyjątkowych można wyróżnić dwie grupy:

1. Dane wejściowe nie spełniają wymagań programu
2. Nie jest możliwy podział grafu na daną liczbę części przy danej dokładności

Poniżej przedstawiono przykłady nieprawidłowych danych wejściowych oraz sytuacji, kiedy nie jest możliwy podział grafu i odpowiadające im komunikaty.

Błąd danych wejściowych	Komunikat
Plik w nieodpowiednim formacie (innym niż tekstowy)	Input file format should be text
Błąd pliku wejściowego (np. wskaźnik z linijki 5 ma większą wartość niż liczba elementów w linijce 4)	Unable to map a graph from input file
Nieprawidłowe parametry wywołania programu (np. ujemna dokładność), nieistniejące flagi	Invalid parameters

We wszystkich powyższych przypadkach program przerywa działanie i po wypisaniu komunikatu wyświetla krótką instrukcję:

Program parameters in order of entering:

Mandatory:

- input_file in text format

Optional:

- number_of_parts (positive integer)

- margin (non-negative integer)

- format (-t for text, -b for binary)

In case of lack of optional parameters program runs with:

number_of_parts = 2, margin = 10, format = -t.

Brak możliwości podziału grafu może wynikać z przesłanek matematycznych. Przykładowo, jeśli graf składa się z 9 wierzchołków, a parametry podziału to 2 części i dokładność 10%, to liczba wierzchołków w otrzymanych częściach będzie się różniła co najmniej o 1, co stanowi więcej niż 10%. Innym przykładem może być polecenie podzielenia grafu o jednym wierzchołku na 2 części z dokładnością 10%. W każdej takiej sytuacji pojawia się komunikat:

Unable to carry out partitioning with following parameters:

[lista parametrów].

Jeśli plik wejściowy zawiera więcej niż jeden graf, program dzieli po kolei każdy z zawartych w pliku grafów zgodnie z podanymi przez użytkownika parametrami. Jeśli dla któregoś z grafów podział nie jest możliwy, program wyświetla powyższy komunikat i przechodzi do kolejnego grafu. Jeśli nie jest możliwy podział żadnego grafu z pliku, program wyświetla ten komunikat dla każdego grafu i kończy działanie.