

# Dokumentacja funkcjonalna

Jan Michorek, Anastasiia Prodius

March 27, 2025

## Contents

<b>1</b>	<b>Wstep</b>	<b>1</b>
<b>2</b>	<b>Technologie</b>	<b>1</b>
<b>3</b>	<b>Architektura systemu</b>	<b>1</b>
<b>4</b>	<b>Struktura kodu</b>	<b>2</b>
<b>5</b>	<b>Przykładowy kod</b>	<b>2</b>
<b>6</b>	<b>Użycie</b>	<b>2</b>
<b>7</b>	<b>Podsumowanie</b>	<b>3</b>

## 1 Wstep

Nasza aplikacja pozwala użytkownikowi wczytywać grafy z pliku o rozszerzeniu csrrg, a następnie dzielić graf na dowolna liczbe części.

## 2 Technologie

Aplikacja została zbudowana przy użyciu następujących technologii:

- Język programowania: C
- Biblioteki: To be defined

## 3 Architektura systemu

Aplikacja składa sie z modułów:

- Moduł generowania grafów
- Moduł analizy grafów
- Moduł wczytywania grafów z pliku

## 4 Struktura kodu

Kod źródłowy podzielony jest na następujące pliki i katalogi:

- src/
  - graph.c
  - csr\_parser.c
- lib/
  - graph.h
  - csr\_parser.h
- output/

## 5 Przykładowy kod

Do przechowywania grafów, służą tak zdefiniowane struktury:

Listing 1: struktura grafu

```
1 typedef struct Node {
2     int id; // identyfikator wierszcholka, 0, 1, ..., n-1
3     int deg; // stopien, czyli liczba krawedzi wierszcholka z innymi
4     struct Node **links; // wskaznik na sasiada
5 } *Node;
6
7 typedef struct Graph {
8     int n;
9     int **mat; // macierz s siedstwa
10    Node *nodes; // wskaznik na wierszcholek 0 w grafie
11
12    // informacje uzyskane z pliku
13    int n_max; // max liczba wierszchokow w wierszu
14    int n_ind[]; // indkesy kolumn w ktorych s wierszcholki
15    int fir_val[]; // wartosci pierwszych wierszcholkow w kazdym rzedzie
16    int edg[]; // tablica krawedzi
17    int edg_ind[]; // indeksy pierwszych wezlow w grupach krawedzi
18 } Graph;
```

W bezpośredniej generacji grafów uczestniczą funkcje podane poniżej:

Listing 2: generacja

```
1 void link_nodes(Node, Node);
2 Node create_Node(int);
3 Graph * graph_init(int);
4 Graph ** part_graphs(int parts, int margin);
```

Zczytywanie z pliku o rozszerzeniu csrrg oraz zaczytywanie z macierzy.

Listing 3: parsowanie i konwertowanie

```
1 void parse_csrrg(FILE *in);
2 void parse_mat(FILE *in);
```

## 6 Użycie

Aplikacja jest uruchamiana z poziomu terminala, z parametrami podanymi w linii poleceń.

#### Listing 4: Przykład uruchomienia programu w terminalu

```
1 ./partg graph.csrrg 3 20
```

gdzie

- **3** – liczba części, na które chcemy podzielić graf.
- **10** – margines, czyli dopuszczalna różnica w liczbie wierzchołków między podgrafami, wyrażona w procentach (wartość w zakresie od 0 do 100%).

#### Skrajne przypadki marginesu:

- **Margines 0%:** Oznacza, że wszystkie podgrafy muszą mieć dokładnie tę samą liczbę wierzchołków, bez żadnej dopuszczalnej różnicy.
- **Margines 100%:** Oznacza, że dopuszczalna różnica w liczbie wierzchołków między podgrafami może wynosić aż 100% całkowitej liczby wierzchołków grafu, co daje pełną swobodę przy podziale (np. jeden podgraf może zawierać wszystkie wierzchołki, a inne podgrafy mogą być puste).

Jeśli nie podano tych parametrów, domyślnie graf będzie podzielony na **dwie części** z **marginesem 10%**.

## 7 Podsumowanie

Aplikacja w podstawowej wersji zapewni użytkownikowi obsługę plików csrrg, i konwertowania sposobów zapisu grafów.