

# Dokumentacja funkcjonalna

Jan Michorek, Anastasiia Prodius

March 26, 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>Podsumowanie</b>	<b>2</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;
3     int ne;
4     struct Node **links;
5 } *Node;
6
7 typedef struct Graph {
8     GraphType type;
9     int n;
10    Node *nodes;
11 } 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, GraphType type);
```

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 Podsumowanie

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