Aplikacja do konwersji reprezentacji grafu oraz jego wizualizacji

Programowanie zespołowe – projekt indywidualny

*Miłosz Jabłoński*

1. **Wprowadzenie**

Niniejszy dokument ma za zadanie zaprezentować funkcje, zastosowanie oraz techniczne aspekty tej aplikacji konwersji reprezentacji grafu oraz jego wizualizacji, która umożliwia przekształcanie danych grafowych pomiędzy różnymi reprezentacjami oraz wizualizację tych danych w formie czytelnej dla użytkownika.

* 1. **Cel i Zastosowanie**

Aplikacja została stworzona z myślą o ułatwieniu analizy, prezentacji i zrozumienia danych grafowych. Grafy są powszechnie wykorzystywane w dziedzinach takich jak sieci społecznościowe, logistyka, nauka o danych, biologia obliczeniowa i wiele innych. Różne reprezentacje grafów mają swoje zalety i ograniczenia, dlatego aplikacja pozwala na konwersję pomiędzy nimi, umożliwiając użytkownikom korzystanie z odpowiedniego formatu w zależności od ich potrzeb.

* 1. **Funkcje Główne**
* Konwersja Reprezentacji Grafu: Centralną funkcją aplikacji jest możliwość konwersji pomiędzy różnymi reprezentacjami grafów. Użytkownicy mogą zmieniać między listą sąsiedztwa, macierzą sąsiedztwa oraz listą krawędzi, zapewniając swobodę manipulacji danymi grafowymi.
* Wizualizacja Grafu: Wizualizacja jest nieodzowna dla zrozumienia złożonych struktur grafowych. Nasza aplikacja pozwala na interaktywną wizualizację grafów, pozwalając użytkownikom eksplorować grafy poprzez dostosowywanie układu. Elastyczność wizualizacji umożliwia ukazanie ukrytych wzorców i informacji. Użytkownicy mogą dostosowywać układ wizualizacji, przesuwać wierzchołki oraz przybliżać/oddalać widok, aby dokładniej przyjrzeć się strukturze grafu.

1. **Docelowy użytkownik i Korzyści**

**2.1 Użytkownik**

Aplikacja do konwersji reprezentacji grafu oraz jego wizualizacji została zaprojektowana z myślą o różnorodnych grupach użytkowników zainteresowanych analizą danych grafowych. Należą do nich:

* Badacze Danych: Osoby analizujące złożone sieci społecznościowe, struktury sieci biznesowych czy relacje naukowe, które potrzebują skoncentrować się na analizie danych grafowych w dogodnej reprezentacji.
* Programiści: Twórcy aplikacji i platform wykorzystujący struktury grafowe, którzy mogą chcieć przekształcić dane grafowe między różnymi formatami w celu optymalizacji wydajności.
* Studenci i Nauczyciele: Osoby uczące się i nauczyciele akademicy w dziedzinach związanych z analizą danych, którzy potrzebują narzędzia do wizualizacji i zrozumienia struktur grafowych w sposób edukacyjny.

**2.2 Korzyści**

Można wyróżnić kilka korzyści płynących z korzystania z aplikacji:

* Zwiększona Produktywność: Użytkownicy oszczędzają czas dzięki automatycznym narzędziom konwersji, zamiast ręcznie przepisywać dane między reprezentacjami, co pozwala na efektywniejszą pracę nad analizą danych.
* Wizualizacja Intuicyjna i Wzbogacona: Aplikacja dostarcza interaktywne i estetyczne wizualizacje grafów, co umożliwia użytkownikom lepsze zrozumienie związków i struktury danych.
* Dokładniejsza Analiza: Przekształcanie reprezentacji grafów pozwala na bardziej skoncentrowaną analizę różnych aspektów danych, ujawniając subtelne detale i wzorce, które mogą ujść uwadze w innych formach reprezentacji.
* Wsparcie w Procesie Decyzyjnym: Analitycy mogą wykorzystać narzędzia w celu lepszego zrozumienia skomplikowanych sieci decyzyjnych, co prowadzi do lepszych i bardziej przemyślanych decyzji.
* Dywersyfikacja Edukacji: Studenci i nauczyciele mogą korzystać z aplikacji jako wzbogacającego elementu edukacji, pozwoli to na zwiększenie angażowania się w tematykę analizy danych grafowych.
* Skalowalność i Wieloaspektowość: Aplikacja jest elastyczna i skalowalna, dzięki czemu może być używana zarówno w analizie małych jak i olbrzymich struktur grafowych oraz może być dostosowana do różnorodnych scenariuszy zastosowań.

1. **Architektura**

Architektura klient-serwer jest fundamentem naszej aplikacji do konwersji reprezentacji grafu oraz jego wizualizacji. Ten model architektoniczny opiera się na podziale funkcjonalności między dwie główne komponenty: klienta, który jest interfejsem użytkownika, oraz serwera, który obsługuje logikę aplikacyjną, przetwarzanie danych oraz komunikację.

* **Klient**
* Klient to interfejs użytkownika, który dostępny jest w przeglądarce internetowej.
* Interaktywny interfejs pozwala użytkownikowi na interakcję z aplikacją poprzez klikanie, przeciąganie i inne działania.
* Klient wyświetla dane, prezentuje wizualizacje i pozwala na wybór operacji, takich jak konwersja reprezentacji grafu czy personalizacja wyglądu wizualizacji.
* Odpowiada za przetwarzanie pewnych zadań logicznych po stronie użytkownika, takich jak odpowiedź na akcje użytkownika.
* **Serwer**
* Serwer to komponent, który zarządza głównymi aspektami logiki aplikacji.
* Odpowiada za przetwarzanie zadań związanych z konwersją reprezentacji grafu, analizą danych oraz generowaniem danych dla wizualizacji.
* Oferuje interfejs API, który umożliwia komunikację między klientem a serwerem poprzez protokół HTTP.

1. **Wykorzystane technologie**

**4.1. Front-end**

* **Język programowania - Javascript**

Wybór języka programowania stanowi kluczową decyzję w procesie tworzenia aplikacji, a w naszym przypadku JavaScript wydaje się być idealnym wyborem ze względu na jego wszechstronność i powszechność. JavaScript jest obecnie jednym z najbardziej używanych języków w dziedzinie tworzenia aplikacji internetowych. Dzięki JavaScriptowi, nasza aplikacja jest dostępna, elastyczna i reaguje na potrzeby użytkowników w czasie rzeczywistym.

* **Framework – React**

Decyzja o wyborze frameworka React wynika z jego zaawansowanej architektury i szerokiego zastosowania w branży tworzenia aplikacji internetowych. Jego modułowy charakter, wydajność dzięki wirtualnemu DOM oraz deklaratywna składnia umożliwiają łatwe tworzenie interaktywnych i responsywnych interfejsów. Bogaty ekosystem, aktywna społeczność oraz narzędzia optymalizacyjne stanowią dodatkowe argumenty, które wspierają tą decyzję. React doskonale współgra z celem projektu, umożliwiając tworzenie zaawansowanych wizualizacji i interakcji z danymi grafowymi.

* **Biblioteki**
* **axios** to narzędzie do wykonywania asynchronicznych żądań HTTP. Dzięki 'axios' frontend może komunikować się z backendem, przesyłając żądania i odbierając odpowiedzi, co pozwala na dynamiczną wymianę danych.
* **MUI\Material-UI** to zestaw gotowych komponentów i stylów bazujących na projekcie Material Design od Google. Używając tych komponentów, możemy tworzyć intuicyjny, spójny i atrakcyjny interfejs użytkownika, co przekłada się na lepszą użyteczność aplikacji.
* **react-graph-vis** to biblioteka która dostarcza narzędzi do tworzenia interaktywnych wizualizacji grafów. React-graph-vis umożliwia dostosowywanie wyglądu wizualizacji, umiejscowienie wierzchołków i łatwą interakcję użytkownika.

**4.2 Back-end**

* **Język programowania – Python**

Język Python jest podstawową technologię backendową aplikacji ze względu na jego czytelność, elastyczność i bogactwo bibliotek. Python oferuje klarowną składnię, co ułatwia zrozumienie kodu i przyspiesza proces tworzenia. Jego ogromna społeczność i obecność na rynku data science sprawiają, że jest idealnym wyborem do pracy z danymi grafowymi. Dzięki dostępowi do bibliotek takich jak igraph, Python umożliwia sprawną analizę i manipulację grafami, doskonale wspierając cel naszej aplikacji.

* **Framework – Flask**

Wybór frameworka Flask jako technologii backendowej wynika z jego minimalistycznego podejścia i elastyczności. Flask oferuje prostą, ale mocną strukturę, co pozwala nam skupić się na budowie logiki aplikacyjnej bez zbędnych skomplikowanych konfiguracji. Jego minimalistyczna natura jest doskonała dla projektu o naszym zakresie, gdzie chcemy skoncentrować się na tworzeniu precyzyjnych funkcji konwersji reprezentacji grafu i generowania wizualizacji.

* **Biblioteki**
* **Igraph** to zaawansowana biblioteka dla języka Python, dedykowana analizie i manipulacji grafami. Wykorzystujemy ją do skomplikowanych operacji konwersji między reprezentacjami grafów oraz do analizy właściwości i struktury grafów.
* **Flask-CORS** to rozszerzenie dla Flask, które umożliwia obsługę żądań międzydomenowych (Cross-Origin Resource Sharing). To istotne dla bezpiecznej komunikacji między frontendem a backendem, zwłaszcza gdy aplikacja działa na różnych domenach.

1. **Instrukcja instalacji i konfiguracji**

* Sklonuj repozytorium z kodem źródłowym z GitHuba

„git clone <https://github.com/311210/Milgraph>”

* Przejdź do katalogu zawierającego kod front-endu

„cd <ścieżka\_do\_katalogu\_frontend>”

* W terminalu wpisz komendę, aby zainstalować zależności projektu

„npm install”

* Po zainstalowaniu zależności, wpisz komendę, aby uruchomić aplikację w trybie deweloperskim

„npm start”

* Przejdź do katalogu zawierającego kod back-endu

„cd <ścieżka\_do\_katalogu\_backend>”

* W celu izolowania zależności projektu, stwórz wirtualne środowisko

„python -m venv venv”

* Następnie, aktywuj wirtualne środowisko

„venv\Scripts\activate”

* W aktywnym wirtualnym środowisku, zainstaluj zależności za pomocą komendy

„pip install -r requirements.txt”

* W aktywnym wirtualnym środowisku, wpisz komendę, aby uruchomić serwer Flask

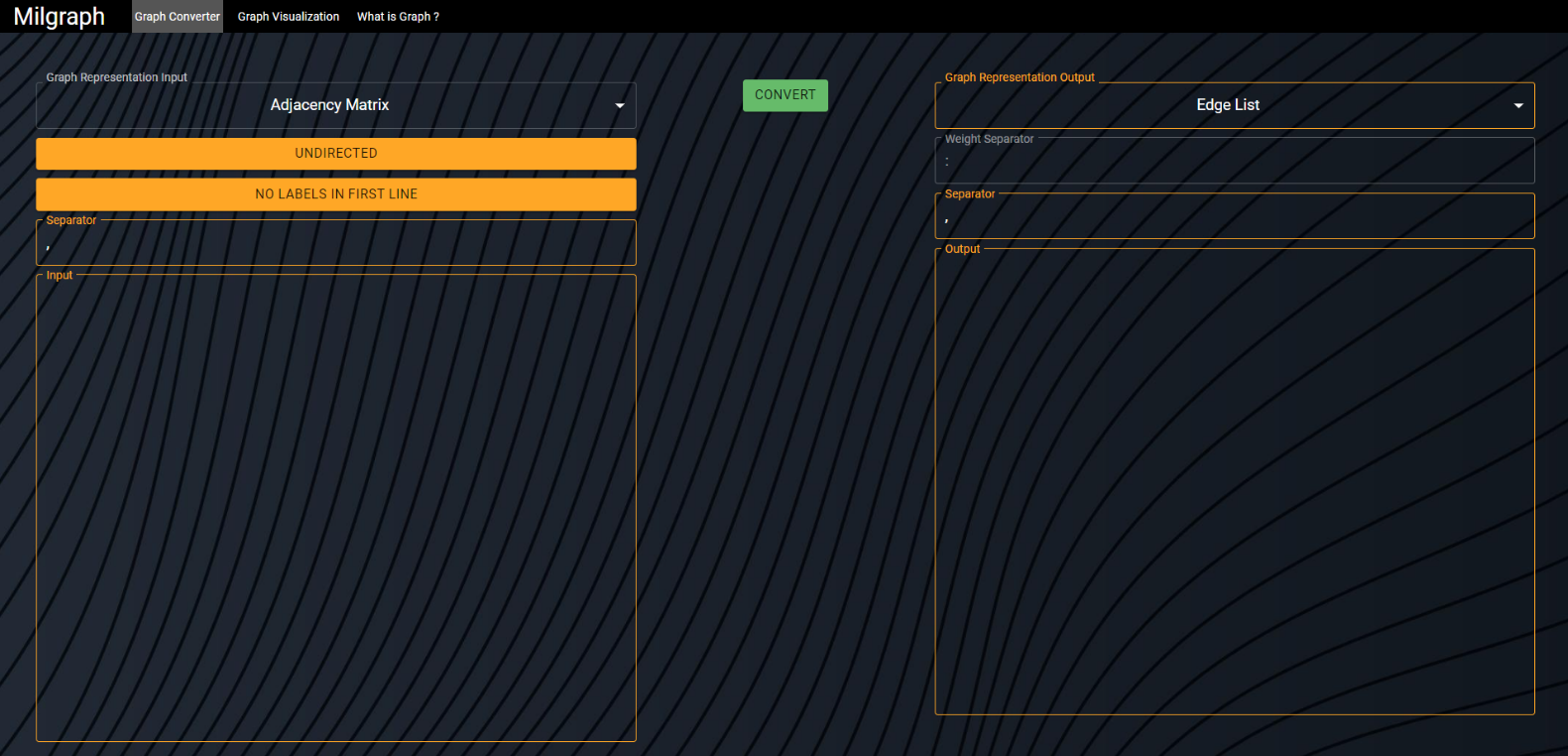
„python server.py”

Aplikacja będzie dostępna pod adresem http://localhost:3000 w przeglądarce.

Serwer będzie dostępny pod adresem http://localhost:5000.

1. **Przewodnik użytkownika**
   1. **Konwersja reprezentacji grafu**

Aplikacja do konwersji reprezentacji grafu umożliwia łatwe przekształcanie danych grafowych pomiędzy różnymi formami reprezentacji, takimi jak lista sąsiedztwa, macierz sąsiedztwa i lista krawędzi. Poniżej znajdziesz kroki, jak korzystać z tej aplikacji:



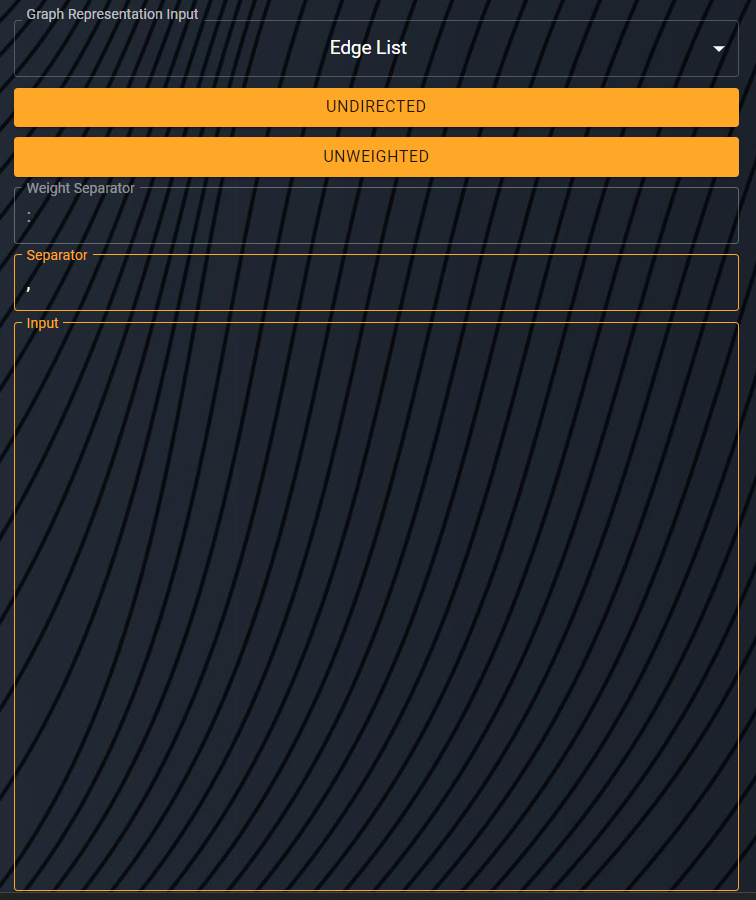
* Wybór Reprezentacji Wejściowej Grafu:

Na górnym panelu aplikacji po lewej stronie znajdziesz rozwijaną listę "Graph Representation Input". Wybierz jedną z trzech opcji reprezentacji grafu: "Edge List" (lista krawędzi), "Adjacency Matrix" (macierz sąsiedztwa) lub "Adjacency List" (lista sąsiedztwa). Wybór ten określa, w jakim formacie podasz dane wejściowe grafu.

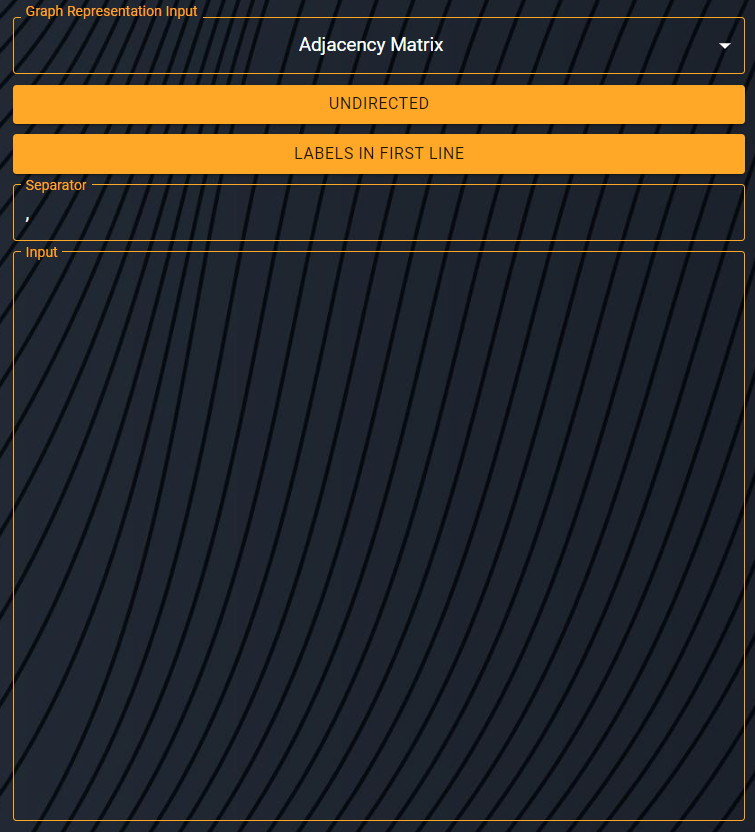
* Ustawienia Konwersji:

Po lewej stronie znajduje się panel do wprowadzania i ustawiania danych wejściowych. Zależnie od wybranej reprezentacji wejściowej, aplikacja umożliwi Ci dostosowanie ustawień konwersji:

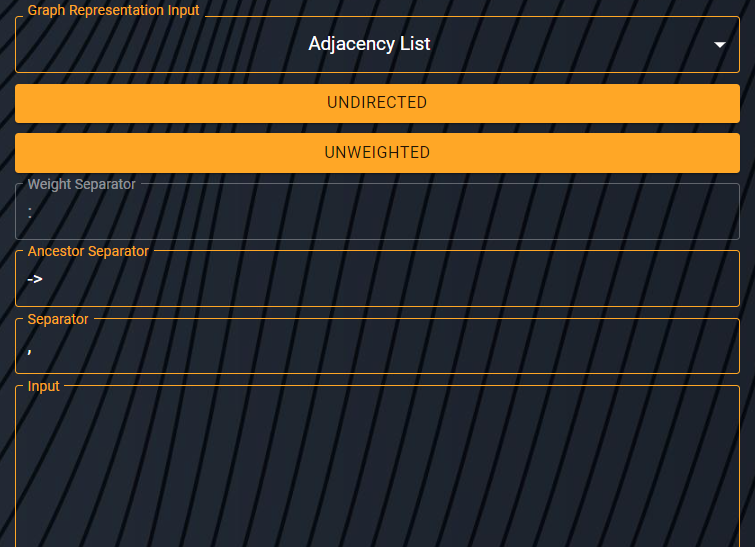
* Edge List (Lista krawędzi)



* Directed/Undirected: Wybierz "Directed", aby oznaczyć graf jako kierunkowy, lub "Undirected" dla grafu nieskierowanego.
* Weighted/ Unweighted: Wybierz "Weighted", aby umożliwić konwersję wag, lub "Unweighted" by graf nie był ważony.
* Weight Separator: Jeśli graf jest ważony pole tekstowe staje się aktywne, możesz określić separator oddzielający wagę od krawędzi.
* Separator (Separator): Pozwala określić separator pomiędzy przodkiem a potomkiem w reprezentacji.
* Adjacency Matrix (Macierz sąsiedztwa)



* Labels in first line/No labels in first line: Pole do wyboru czy w naszej reprezentacji w pierwszej linii znajdują się nazwy poszczególnych wierzchołków grafu.
* Separator (Separator): Pozwala określić separator pomiędzy liczbami w macierzy
* Directed/Undirected: Wybierz "Directed", aby oznaczyć graf jako kierunkowy, lub "Undirected" dla grafu nieskierowanego.
* Adjacency List (Lista sąsiedztwa)



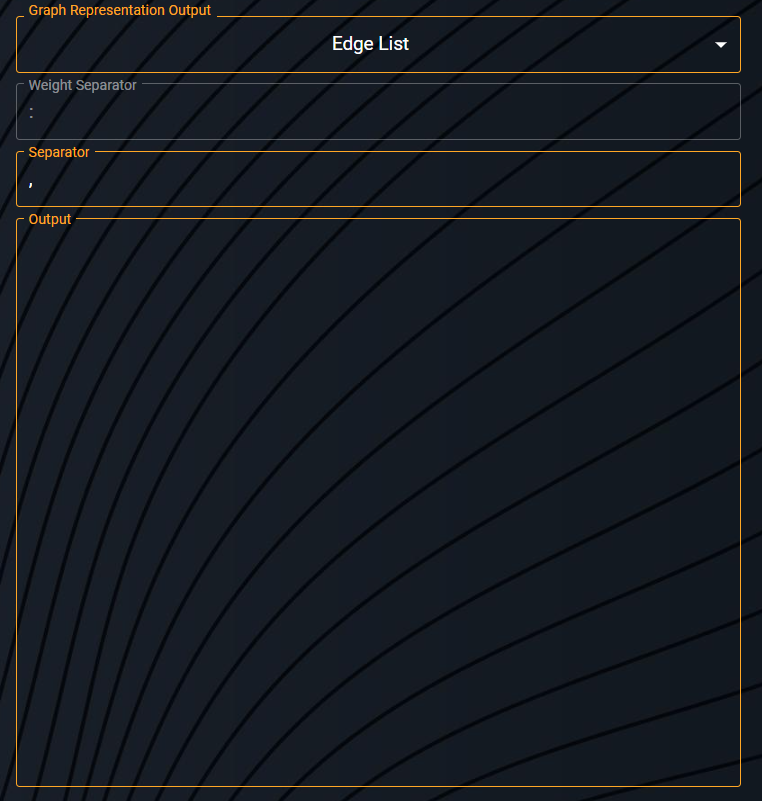
* Weighted/ Unweighted: Wybierz "Weighted", aby umożliwić konwersję wag, lub "Unweighted" by graf nie był ważony.
* Ancestor separator (Separator przodka): Pozwala określić separator pomiędzy przodkiem a potomkami w reprezentacji.
* Separator (Separator): Pozwala określić separator pomiędzy potomkami w reprezentacji.
* Directed/Undirected: Wybierz "Directed", aby oznaczyć graf jako kierunkowy, lub "Undirected" dla grafu nieskierowanego.
* Wprowadzenie Danych Wejściowych:

W polu "Input" wprowadź dane grafowe zgodnie z wybraną reprezentacją. Upewnij się, że oddzielasz wierzchołki/krawędzie i wagi (jeśli występują) odpowiednim separatorem. W przypadku macierzy sąsiedztwa, wprowadź wiersze macierzy, oddzielając elementy separatorem.

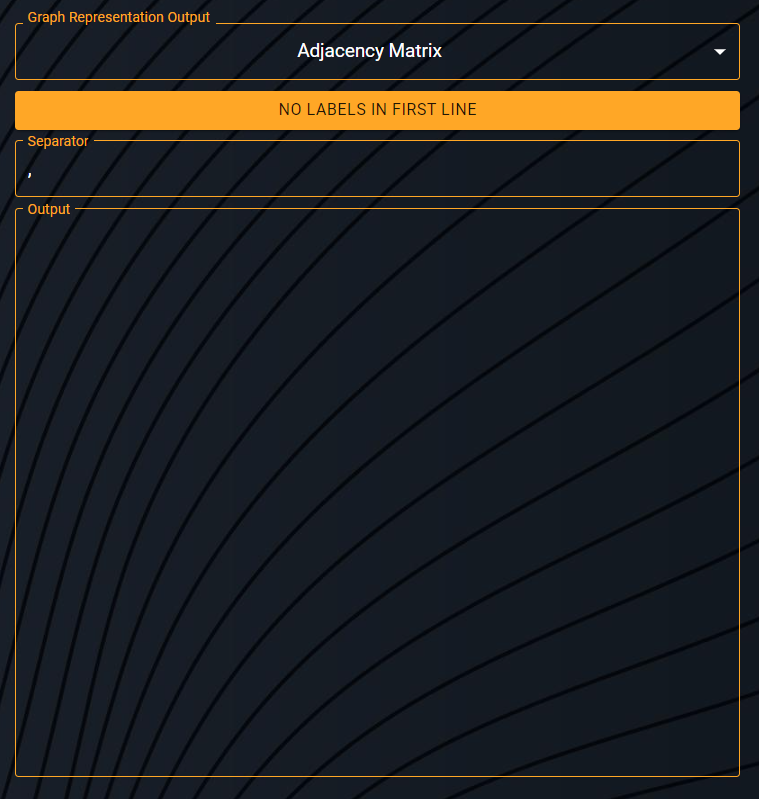
* Wybór Reprezentacji Wyjściowej Grafu:

Na panelu po prawej stronie, znajdziesz rozwijaną listę "Graph Representation Output". Wybierz jeden z trzech formatów reprezentacji grafu: "Edge List", "Adjacency Matrix" lub "Adjacency List". Wybór ten określa, w jakim formacie chcesz otrzymać wynik konwersji.

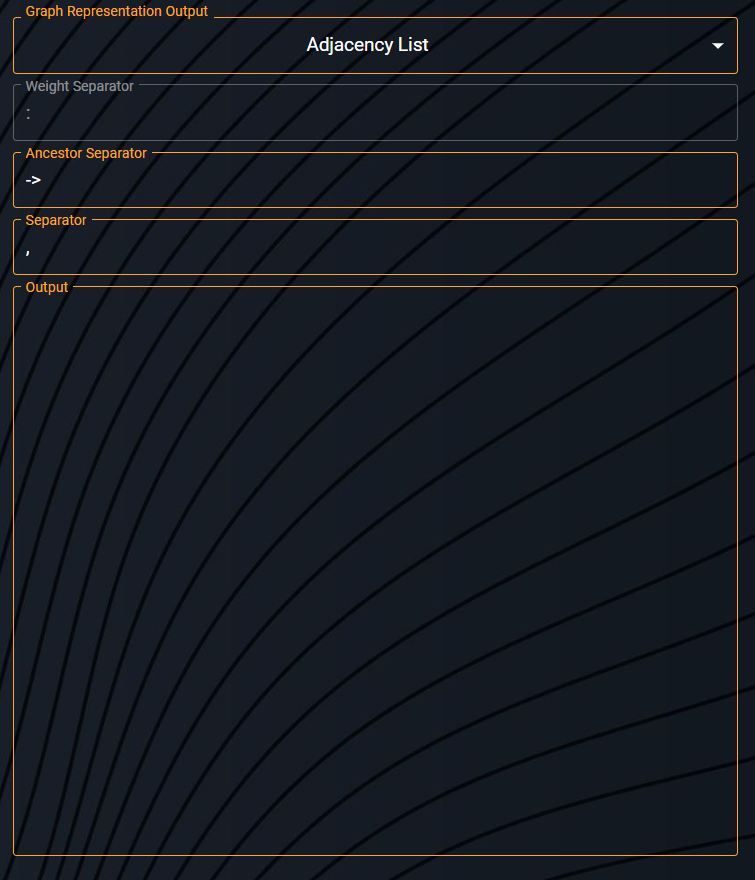
* Edge List (Lista krawędzi)



* Weight Separator: Jeśli graf jest ważony pole tekstowe staje się aktywne, możesz określić separator oddzielający wagę od krawędzi dla danych wyjściowych.
* Separator (Separator): Pozwala określić separator pomiędzy przodkiem a potomkiem w reprezentacji.
* Adjacency Matrix (Macierz sąsiedztwa)



* Labels in first line/No labels in first line: Pole do wyboru czy w reprezentacji wyjściowej w pierwszej linii mają znajdować się nazwy poszczególnych wierzchołków grafu.
* Separator (Separator): Pozwala określić separator pomiędzy liczbami w macierzy w danych wyjściowych.
* Adjacency List (Lista sąsiedztwa)

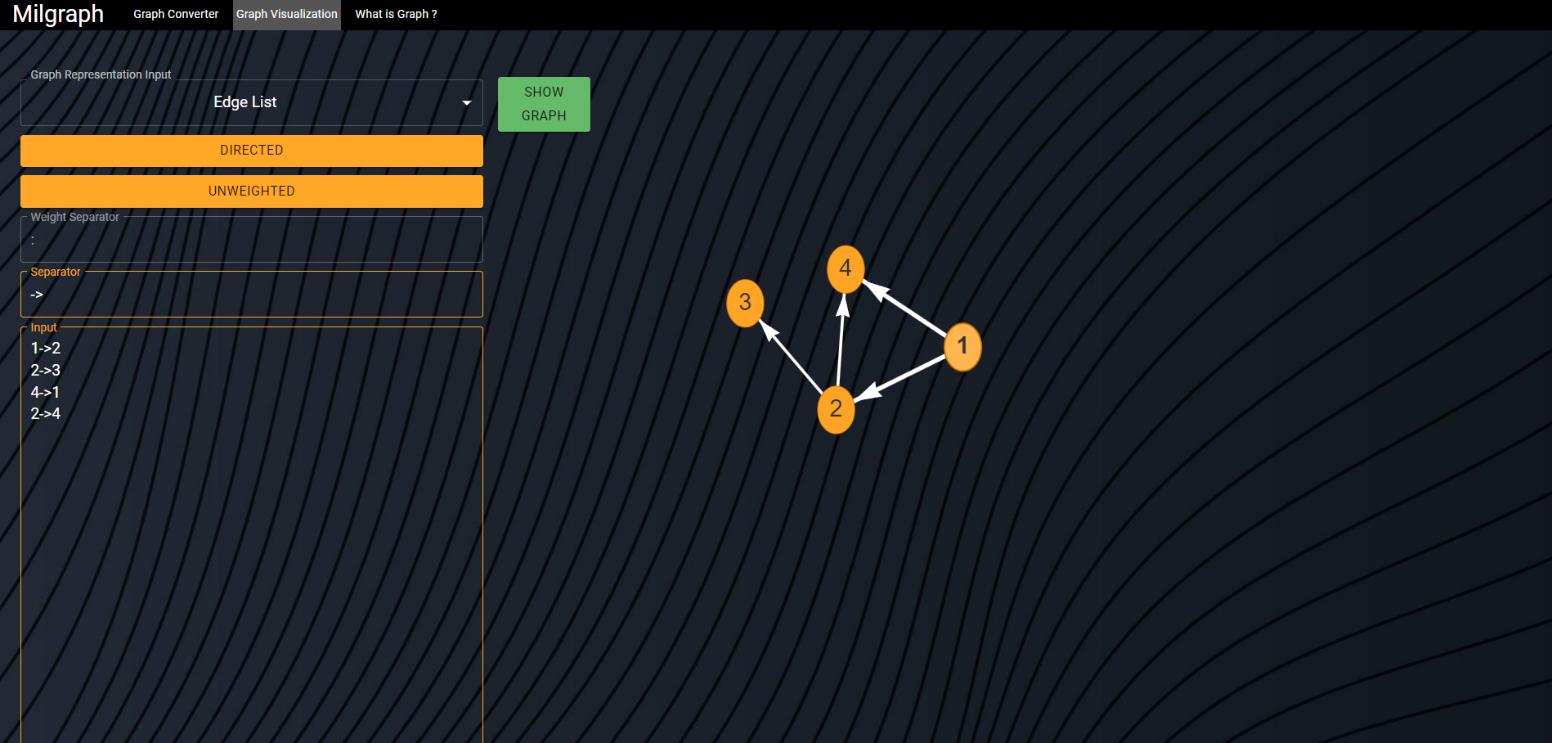


* Weight Separator: Jeśli graf jest ważony pole tekstowe staje się aktywne, możesz określić separator oddzielający wagę od krawędzi dla danych wyjściowych.
* Ancestor separator (Separator przodka): Pozwala określić separator pomiędzy przodkiem a potomkami w reprezentacji.
* Separator (Separator): Pozwala określić separator pomiędzy potomkami w reprezentacji.
* Konwersja Grafu:

Po wprowadzeniu wszystkich danych i dostosowaniu ustawień, kliknij przycisk "Convert". Aplikacja wyśle dane na serwer, gdzie zostanie przeprowadzona konwersja. Wynik konwersji zostanie wyświetlony w polu "Output".

* 1. **Wizualizacja grafu**

Aplikacja do wizualizacji grafu umożliwia łatwe przekształcanie danych grafowych takich jak lista sąsiedztwa, macierz sąsiedztwa i lista krawędzi w wizualizacje grafu. Poniżej znajdziesz kroki, jak korzystać z tej aplikacji:



* Wybór Reprezentacji Wejściowej Grafu:

Na górnym panelu aplikacji po lewej stronie znajdziesz rozwijaną listę "Graph Representation Input". Wybierz jedną z trzech opcji reprezentacji grafu: "Edge List" (lista krawędzi), "Adjacency Matrix" (macierz sąsiedztwa) lub "Adjacency List" (lista sąsiedztwa). Wybór ten określa, w jakim formacie podasz dane wejściowe grafu.

* Ustawienia Wizualizacji:

Po lewej stronie znajduje się panel do wprowadzania i ustawiania danych wejściowych. Zależnie od wybranej reprezentacji wejściowej, aplikacja umożliwi Ci dostosowanie ustawień konwersji:

* Edge List (Lista krawędzi)
* Directed/Undirected: Wybierz "Directed", aby oznaczyć graf jako kierunkowy, lub "Undirected" dla grafu nieskierowanego.
* Weighted/ Unweighted: Wybierz "Weighted", aby umożliwić konwersję wag, lub "Unweighted" by graf nie był ważony.
* Weight Separator: Jeśli graf jest ważony pole tekstowe staje się aktywne, możesz określić separator oddzielający wagę od krawędzi.
* Separator (Separator): Pozwala określić separator pomiędzy przodkiem a potomkiem w reprezentacji.
* Adjacency Matrix (Macierz sąsiedztwa)
* Labels in first line/No labels in first line: Pole do wyboru czy w naszej reprezentacji w pierwszej linii znajdują się nazwy poszczególnych wierzchołków grafu.
* Separator (Separator): Pozwala określić separator pomiędzy liczbami w macierzy
* Directed/Undirected: Wybierz "Directed", aby oznaczyć graf jako kierunkowy, lub "Undirected" dla grafu nieskierowanego.
* Adjacency List (Lista sąsiedztwa)
* Weighted/ Unweighted: Wybierz "Weighted", aby umożliwić konwersję wag, lub "Unweighted" by graf nie był ważony.
* Ancestor separator (Separator przodka): Pozwala określić separator pomiędzy przodkiem a potomkami w reprezentacji.
* Separator (Separator): Pozwala określić separator pomiędzy potomkami w reprezentacji.
* Directed/Undirected: Wybierz "Directed", aby oznaczyć graf jako kierunkowy, lub "Undirected" dla grafu nieskierowanego.
* Wprowadzenie Danych Wejściowych:

W polu "Input" wprowadź dane grafowe zgodnie z wybraną reprezentacją. Upewnij się, że oddzielasz wierzchołki/krawędzie i wagi (jeśli występują) odpowiednim separatorem. W przypadku macierzy sąsiedztwa, wprowadź wiersze macierzy, oddzielając elementy separatorem.

* Wizualizacja Grafu:

Po wprowadzeniu wszystkich danych i dostosowaniu ustawień, kliknij przycisk "Show Graph". Aplikacja wyśle dane na serwer, gdzie zostanie przeprowadzona konwersja i przygotowany wykres grafu. Wykres grafu pojawi się w sekcji wizualizacji. Po wygenerowaniu wykresu grafu, możesz z niego korzystać w sposób interaktywny. Przesuwaj się po grafie, przybliżaj, oddalaj i klikaj na wierzchołki aby je przesunąć.