**DOKUMENTACJA PROJKETU**

***Implementacja algorytmu Dijkstry***

Pakiet obliczeniowy MATLAB i jego zastosowania

Adrianna Pytel, Emilia Rutkowska

1. **Wstęp.**

Algorytm Dijkstry, nazywany także algorytmem najkrótszych ścieżek, został wynaleziony przeszło pół wieku temu, przez holenderskiego informatyka Edsger'a W. Dijkstra. Celem algorytmu jest znalezienie najkrótszej ścieżki między zadanymi dwoma wierzchołkami grafu, czy też stworzenie tablicy najkrótszych odległości od zadanego wierzchołka.

1. **Opis działania algorytmu.**

Zaczynamy od wybrania wierzchołka startowego. Będziemy próbowali zbudować tablicę najkrótszych odległości od wierzchołka startowego do wszytskich pozostałych wierzchołków. W każdym kroku algorytmu będziemy dokonywać uaktualnienia informacji w trzech tablicach:

* tablicy najkrótszych odległości od wierzchołka startowego do pozostałych wierzchołków
* tablicy poprzedników, zawierającą informację o wierzchołku który odwiedziliśmy jako ostatni
* tablicy nieodwiedzonych wierzchołków

Na początek uzupełniamy tablicę najkrótszych odległości - przypisując 0 jako odległość z wierzchołka startowego do niego samego oraz nieskończoność jako odległośc do pozostałych wierzchołków. Tablica nieodwiedzonych wierzchołków zawiera wszystkie wierzchołki.

W każdej iteracji algorytmu:

* wybieramy nieodwiedzony wierzchołek (bieźący) o najmniejszej odległości do wierzchołka startowego (na podstawie tablicy odległości)
* dla bieżącego wierzhcołka szukamy jego sąsiadów oraz obliczamy odległość każdego wierzchołka-sąsiada od wierzchołka startowego
* jeśli obliczona odległość jest mniejsza od znanej odległości (zapisanej w tablicy odleglości) to uaktualniamy tablicę odległości
* uaktualniamy tablicę poprzedników oraz tablicę odwiedzonych wierzchołków

1. **Funkcjonalności aplikacji oraz ich użycie.**

W widoku startowym użytkownika wita zwięzła instrukcja obsługii aplikacji, okienko wprowadzania danych oraz przycisk restart. Daną, o którą aplikacja w pierwszej kolejności prosi użytkownika jest rozmiar grafu (ilość budujących graf wierzchołków) mieszczący się w granicach od 1 do 10. W razie wpisania niepoprawnej wartości użytkownik zostaje poinformowany o błędzie i o dopuszczalnym zakresie rozmiaru grafu. . Wprowadzony rozmiar należy zatwierdzić przycieskiem Enter lub klikając w dowolnym miejscu okna aplikacji.

W efekcie pojawia się macierz odległości, do której uzytkownik ma możliwość wprowadzenia nieujemnych wartości wag krawędzi grafu. Macierz automatycznie uzupełnia wartości symetrycznie względem przekątnej diagonalnej. Po uzyskaniu porządanej macierzy odległości, generowanie grafu odbywa się poprzez naciśnięcie przycisku: create graph. Zostaje wyrysowany graf.

Kolejnym elementem interfejsu użytkownika są dwa pola, pozwalające na wprowadzenie wierzchołka startowego i końcowego. Dla podanych wierzchołków istnieje możliwość znalezienia najkrótszej łączącącej ich ścieżki w grafie. W tym miejscu zostało dodane zabezpiecznie, uniemożliwiajace wprowadzenie wierzchołka, który nie istnieje. W przypadku próby wyznaczenia odległości między wierzchołkami, które istnieją w grafie, ale które budują osobne drzewa - również zostaniemy poinformowani o błędzie.   
Gdy wprowadzone zostaną poprawne wartości wierzchołków, najkrótszą ścieżkę - możemy uzyskać poprzez kliknięcie przycisku: Shortest path, tymsamym wywołując algorytmu Dijksty. Na wykresie kolorem zielonym zostanie zaznaczona najkrótsza ścieżka.

Ponadto zostanie wygenerowana macierz najkrotszych odległości między wszystkimi wierzchołkami. Dla lepszej wizualizacji, poszukiwany przez nas najkrótszy dystans zostanie zaznaczony pokolorowaniem na kolor żołty odpowiednich pól macierzy. W przypadku wierzchołków znajdujących się w różnych niepołączonych ze sobą podgrafach grafu, odległość między nimi w tablicy wynosi Inf. Próba znalezienia najkrótszej ściezki między niepołączonymi wierzchołkami zakończy się komunikatem o błędzie. Nie zostanie wypisana obecna w tablicy najkrótszych odległości wartość Inf.

W trakcie działania aplikacji istnieje możliwość modyfikacji wszystkich parametrów, jak również wykonania całościowego resetu doknanych zmian. (o błędzie że zostaje uprzednio wybrana wartosć rozmiaru grafu, więc jeśli chcielibyśmy ją nadpisać tą samą wartością - możemy tego dokonać poprzez 1. zmianę na inną wartość 2. wprowadzenie ponownie oczekiwanej wartosci).

1. **Skrócona instrukcja użycia.**
2. Wprowadź wielkość grafu (od 1 do 10), zatwierdź wybór ENTER (lub odkliknięciem).
3. Uzupełnij macierz wag, oczekiwanymi wagami połączeń, po jednej stronie diagonali. Np. wartość 7 wprowadzona w polu (3,5) oznacza: stwórz krawędź o wadze 7 między wierzchołkami: trzecim i piątym.
4. Zatwierdź wybór macierzy wag kilkając przycisk: Create graph.
5. Spróbuj znaleźć najkrótszą ścieżkę w grafie: wprowadź w startNode wierzchołek początkowy ścieżki, w endNode - wierzchołek końcowy.
6. Zatwierdź wybór klikając przycisk: Shortest path.
7. Modyfikuj wprowadzone parametry lub naciśnij przycisk Reset i wprowadź wartości od początku.
8. **Błędy, ostrzeżenia.**

W tym punkcie zostały zaprezentowne mogące pojawić się w trakcie interakcji z aplikacją wykryte błędy oraz komunikaty.

1. Wpisywanie rozmiaru grafu:
   * Jeżeli wprowadzona wartość rozmiaru grafu jest spoza zakresu [1,10] wyświetli się komunikat o błędzie.
   * Jeżeli zmodyfikujemy rozmiar grafu gdy inny graf jest już stworzony, to poprzedne dane zostaną usunięte
2. Wpisywanie danych do macierzy sąsiedztwa:
   * Jeżeli wprowadzona wartość będzie liczbą ujemną, literą lub znakiem specjalnym pojawi się komunikat o błędnym formacie danych. Wyjątkiem jest litera ‘i’, która zostanie potraktowana jako liczba urojona. W takim przypadku program może nie zadziałać poprawnie.
   * Jeżeli zostanie zmieniona wartość na diagonali macierzy, wyświetli się informacja, że te dane nie mogą ulec zmianie.
   * Dla kolumny nr 10 (jeśli występuje) program wyświetli nazwę 1 zamiast 10.
   * Macierz sąsiedztwa można zmieniać w każdym momencie działania program, bez utraty poprzednich danych, jednak należy pamiętać, aby po zmianie na nowo kliknąć przycisk “Create Graph”.
3. Wprowadzanie wierzchołków w pola startNode/endNode:
   * Jeżeli wybierzemy wierzołek o numerze 0, zostanie wyświetlony komunikat, że taki wierzchołek nie istenieje.
   * Jeżeli wybrane wierzchołki należą do różnych podgrafów, które są ze sobą niespójne, to użytkownik zostanie poinformowany, że takie połączenie wymaga stworzenia spójnego grafu.
4. **Podział pracy.**

Emilia Rutkowska – implementacja algorytmu Dijkstry, stworzenie dokumentacji projektu

Adrianna Pytel – implementacja GUI i reszty programu