

Specyfikacja funkcjonalna programu służącego do analizy grafów.

Filip Sosnowski, Krzysztof Tadeusiak

19.05.2022

Cel projektu

Program służy do analizy grafu wygenerowanego na podstawie podanych przez użytkownika argumentów. Umożliwia znalezienie najkrótszej drogi pomiędzy wybraną parą węzłów, a także oferuje sprawdzenie spójności grafu. Wygenerowane dane są zapisywane do pliku tekstowego. Program potrafi wczytać graf z pliku o ustalonym formacie. Program przystosowany jest do odczytywania i generowania grafów za pomocą plików stworzonych z wcześniejszego projektu napisanego w języku C. Nasze rozwiązanie pozwala zwizualizować graf, a także umożliwia wybór węzłów do wyznaczania ścieżki za pomocą myszki. Program ten potrafi również pokazać najkrótszą ścieżkę między dwoma wybranymi wierzchołkami. W naszym projekcie wykorzystane jest działanie algorytmu Breadth-first Search (BFS) oraz algorytmu Dijkstry.

Teoria

Graf - zbiór wierzchołków, które mogą być połączone krawędziami, w taki sposób, że każda krawędź kończy się i zaczyna w pewnym wierzchołku. Wierzchołki grafu są numerowane. Krawędzie obrazują relacje pomiędzy obiektami.

Graf wagowy - krawędzie mogą mieć przypisaną wagę, tzn. przypisaną wartość liczbową. Określa ona w naszym projekcie długość drogi - im mniejsza waga, tym krótsza droga pomiędzy dwoma wierzchołkami.

Spójność grafu - graf jest spójny, gdy każda para wierzchołków jest połączona ścieżką.

Algorytm BFS - to algorytm, który przeszukuje grafy wszerek. Służyć może do sprawdzenia spójności grafu lub do znalezienia najkrótszej drogi pomiędzy dwoma wybranymi wierzchołkami grafu.

Algorytm Dijkstry - algorytm, który służy do znalezienia na grafie najkrótszej drogi między określonym węzłem a każdym innym. Znajduje również

zastosowanie, gdy należy znaleźć najkrótszą ścieżkę od określonego wierzchołka startowego do wybranego węzła końcowego. Algorytm Dijkstry na wejściu wymaga grafu skierowanego lub nieskierowanego o dodatnich wagach krawędzi.

Dane wejściowe

Program wymaga danych, które opisują parametry tworzonego grafu. Każdy z grafów musi być opisany następującymi parametrami:

- liczba kolumn: na jej podstawie program może utworzyć siatkę o danej ilości wierzchołków znajdujących się kolejno po sobie wzdłuż osi X. Dopuszczalne wartości: $x > 0$, $x \leq 10^6$, $x \times y \leq 10^6$ oraz $x \in C$. Użytkownik może ją podać, wpisując wcześniej podaną, dopuszczalną wartość w pole tekstowe przy symbolu "x";
- liczba wierszy: na jej podstawie program może utworzyć siatkę o danej ilości wierzchołków znajdujących się kolejno po sobie wzdłuż osi Y. Dopuszczalne wartości: $y > 0$, $y \leq 10^6$, $x \times y \leq 10^6$ oraz $y \in C$. Użytkownik może ją podać, wpisując wcześniej podaną, dopuszczalną wartość w pole tekstowe przy symbolu "y";
- minimalna wartość wagi krawędzi: determinuje jaka minimalna waga może pojawić się między wierzchołkami w grafie. Dopuszczalne wartości: $\min \geq 0$, $\min \leq 1$ oraz $\min \in R$. Użytkownik może ją podać, wpisując wcześniej podaną, dopuszczalną wartość w pole tekstowe przy słowie "min";
- maksymalna wartość wagi krawędzi: determinuje jaka maksymalna waga może pojawić się między wierzchołkami w grafie. Dopuszczalne wartości: $\max \geq 0$, $\max \leq 1$ oraz $\max \in R$. Użytkownik może ją podać, wpisując wcześniej podaną, dopuszczalną wartość w pole tekstowe przy słowie "max";
- punkt startowy: determinuje, z którego wierzchołka program ma zacząć znajdować najkrótszą drogę do punktu końcowego. Dopuszczalne wartości: $ps \geq 0$, $ps \leq x \times y - 1$ oraz $ps \in C$. Użytkownik może go ustalić, wpisując wcześniej podaną, dopuszczalną wartość w pole tekstowe przy słowie "ps";
- punkt końcowy: determinuje, do którego wierzchołka program ma wyznaczyć najkrótszą drogę z punktu startowego. Dopuszczalne wartości: $pk \geq 0$, $pk \leq x \times y - 1$ oraz $pk \in C$. Użytkownik może go ustalić, wpisując wcześniej podaną, dopuszczalną wartość w pole tekstowe przy słowie "pk";
- punkt startowy oraz punkt końcowy będzie można ponownie określić na uprzednio wygenerowanym grafie poprzez klikanie na interesujące nas wierzchołki;
- liczba podziału grafu: determinuje, ile grafów otrzymamy z podstawowo wygenerowanego grafu. Dopuszczalne wartości: $n \geq 1$, $n \leq \frac{x \times y}{4}$ oraz $n \in C$. Użytkownik może ją podać, wpisując wcześniej podaną dopuszczalną, wartość w pole tekstowe przy słowie "n";

Dane wyjściowe

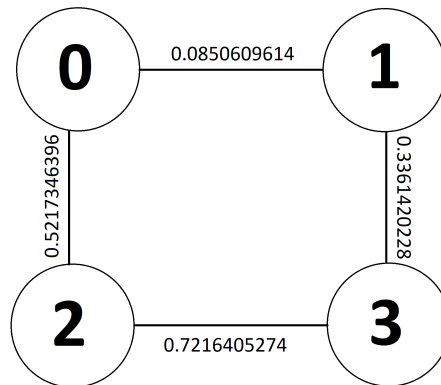
Program ten generuje graf o określonych wymiarach oraz wartościach wag krawędzi oraz zapisuje go do pliku o ustalonym formacie.

Przykładowy plik z grafem:

```
2 2
2: 0.5217346396 1: 0.0850609614
3: 0.3361420228 0: 0.0850609614
0: 0.5217346396 3: 0.7216405274
2: 0.7216405274 1: 0.3361420228
```

W pierwszej linijce określona jest liczba kolumn oraz wierszy. Kolejne wiersze opisują wartości wag krawędzi w rozważanym grafie.

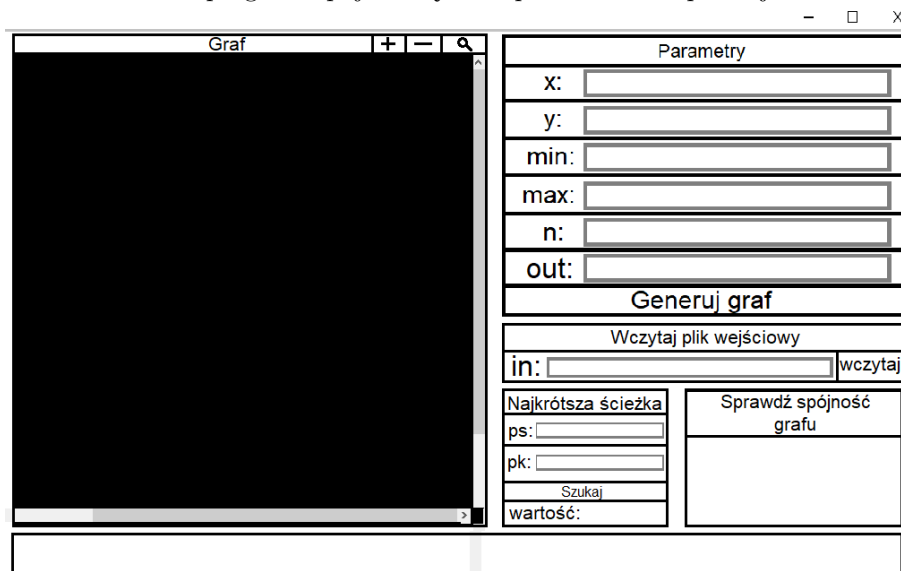
Poniżej załączona jest wizualizacja graficzna wygenerowanego grafu.



Program określi spójność grafu za pomocą algorytmu BFS. Wskaże także najkrótszą możliwą drogę między wybranymi węzłami, korzystając z algorytmu Dijkstry. Obie te informacje zostaną zakomunikowane w strumieniu wyjścia.

Opis graficznego interfejsu użytkownika

Po uruchomieniu programu pojawi się okno przedstawione poniżej.



Widoczne jest sześć paneli:

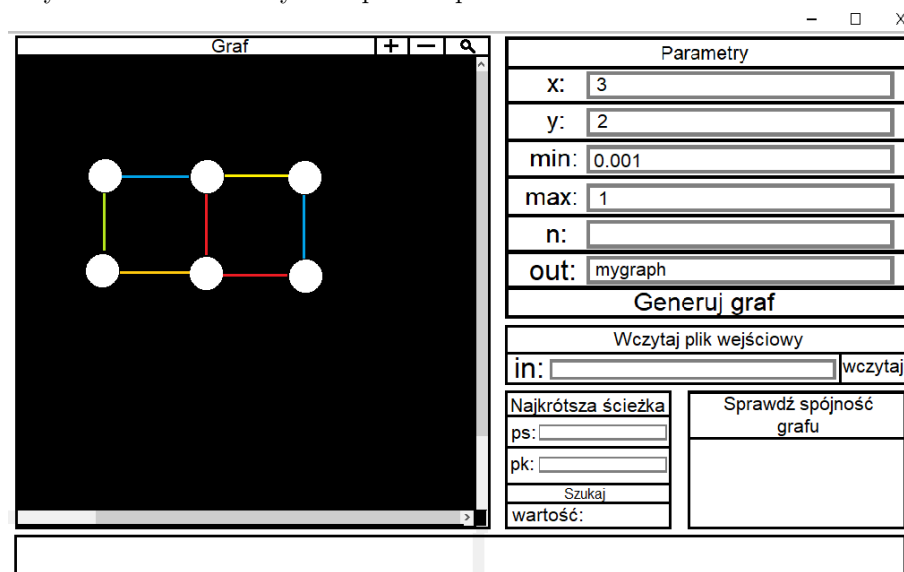
- Parametry - ten panel umożliwia podanie argumentów wejściowych takich jak liczba kolumn (x), liczba wierszy (y), minimalna (min) oraz maksymalna (max) wartość przejścia między węzłami, a także liczba podziału grafu (n) i nazwa pliku wyjściowego (out). Wybrane wartości użytkownik wprowadza w pole tekstowe. By wygenerować graf na podstawie podanych argumentów, należy kliknąć przycisk „Generuj”.
- Graf - w tym panelu zostanie zwizualizowany graf, który został wygenerowany lub wczytany przez użytkownika. Umożliwi wybór węzłów do wyznaczenia najkrótszej ścieżki za pomocą myszki, a także pokaże ją na grafie. Graf zostanie przedstawiony w postaci siatki. Skrzyżowania kratek to węzły, a linie pionowe i poziome to krawędzie. Wagę krawędzi odwzorowane zostaną kolorem, gdzie kolor ciemnoniebieski odpowiada minimalnej wartości, a kolor ciemnoczerwony - maksymalnej wartości. Graf można przybliżać za pomocą przycisku "+" oraz oddalać za pomocą przycisku "-". Do przybliżenia można wykorzystać również przycisk z lupą. By przesuwać się wzdłuż i w szerz grafu można wykorzystać paski przewijania.
- Wczytaj plik wejściowy - panel ten odpowiada za wczytywanie pliku wejściowego o ustalonym formacie z grafem. Nazwę pliku należy wpisać w pole tekstowe obok słowa "in:", a następnie wcisnąć przycisk "wczytaj".
- Najkrótsza ścieżka - umożliwia wyszukanie najkrótszej ścieżki pomiędzy dwoma wybranymi punktami - punktem startowym (ps) oraz punktem koń-

cowym (pk). Odpowiednie numery węzłów należy wpisać w pola tekstowe umieszczone przy "ps" oraz "pk". By zatwierdzić, trzeba wcisnąć przycisk "Szukaj". Jeśli wprowadzone dane są poprawne, wartość najkrótszej ścieżki pomiędzy tymi węzłami zostanie wypisana w polu "Wartość". Jednocześnie w panelu graf ta ścieżka zostanie zaznaczona kolorem białym.

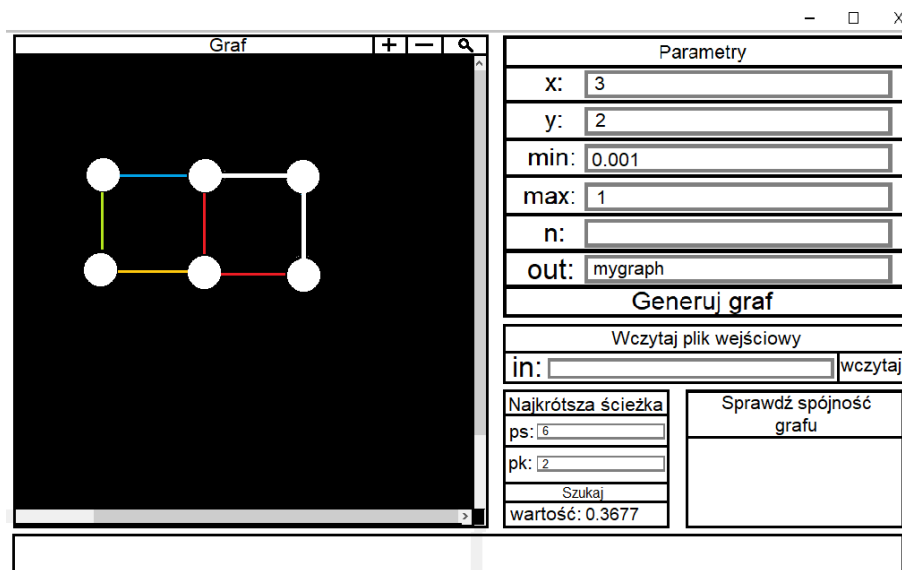
- Sprawdź spójność grafu - na podstawie algorytmu BFS, panel ten pokaże czy rozważany graf jest spójny. By się tego dowiedzieć, należy wcisnąć przycisk "Sprawdź spójność grafu". Informacja o spójności zostanie wypisana w polu pod przyciskiem.
- Komunikaty - panel umieszczony na samym dole okna programu jest odpowiedzialny za wypisywanie błędów oraz informacji.

Przykład działania interfejsu

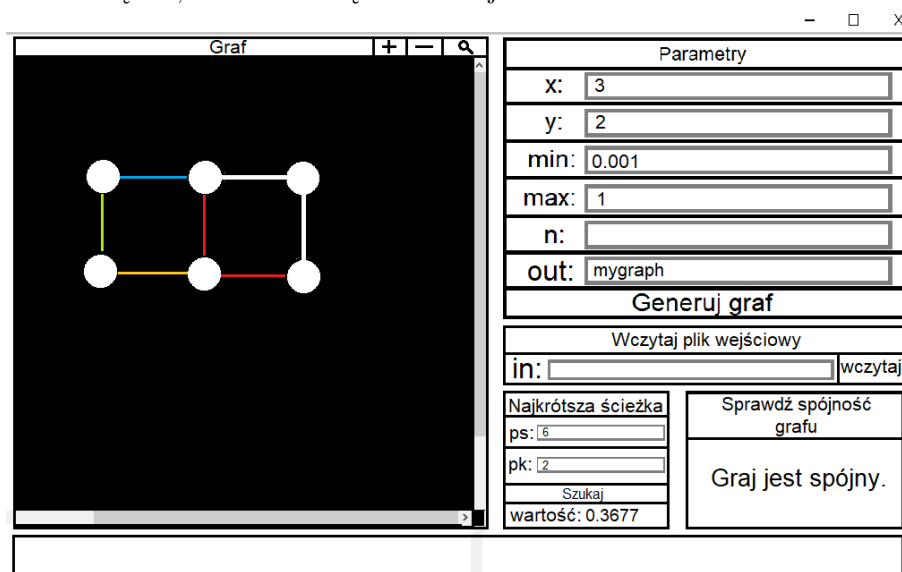
Przykład działania w trybie zapisu do pliku:



Po podaniu w panelu "Parametry" wybranych wartości oraz zatwierdzeniu ich przyciskiem "Generuj", w panelu "Graf" został wygenerowany graf o odpowiednich wymiarach oraz wagach krawędzi przejścia. Graf został zapisany do pliku wyjściowego "mygraph".

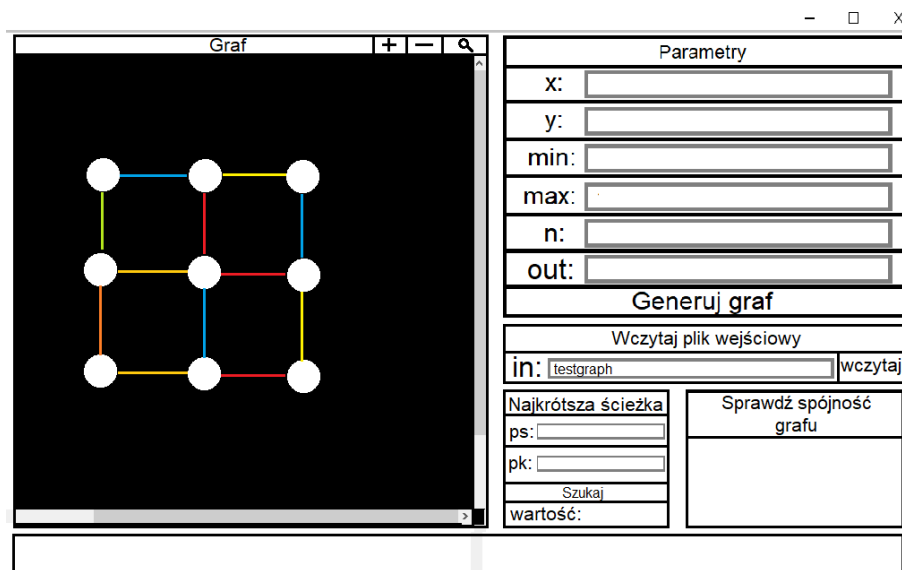


Po podaniu w panelu "Najkrótsza ścieżka" odpowiednich wartości punktu startowego oraz końcowego oraz zatwierdzeniu ich przyciskiem "Szukaj", została wypisana wartość długości tej ścieżki, a w panelu "Graf" kolorem białym zaznaczono krawędzie, które wchodzą w skład tej ścieżki.



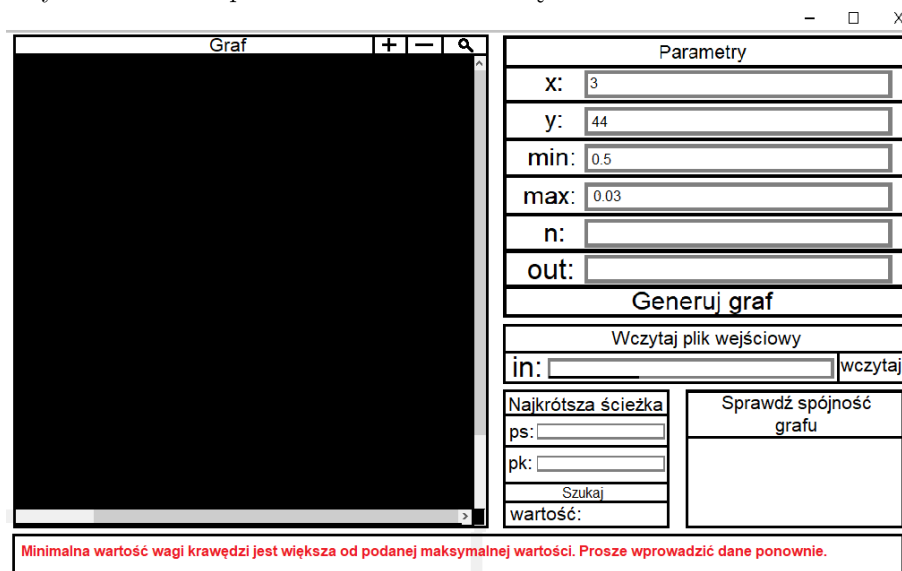
Po wciśnięciu przycisku "Sprawdź spójność grafu" zostaje poniżej tego przycisku wypisana informacja dotycząca spójności grafu, w tym przypadku "Graf jest spójny".

Przykład działania w trybie odczytu z pliku:



Po wpisaniu w panelu "Wczytaj plik wejściowy" nazwy pliku wejściowego z grafem o (odpowiednim formacie) oraz zatwierdzeniu go przyciskiem "Wczytaj", graf ten zostanie zwizualizowany w panelu "Graf". Sprawdzenie najkrótszej ścieżki oraz spójności grafu działa analogicznie do przypadków przedstawionych wyżej.

Przykład działania panelu z komunikatami błędów:



Komunikaty błędów

Dane numeryczne akceptowalne są jedynie w systemie dziesiętnym rzeczywistym. Funkcję separatora dziesiętnego będzie pełnić kropka („.”). Program nie obsługuje podanych przez użytkownika liczb ujemnych. Wyświetla on wtedy odpowiedni komunikat błędu oraz prosi użytkownika o sprawdzenie danych i ich ponowne wprowadzenie. Program pobiera od użytkownika wartość 0 tylko i wyłącznie dla danych "minimalnej wartości wag krawędzi" oraz "maksymalnej wartości wag krawędzi". Dla innych danych wyświetla komunikat błędu i prosi o ponowne ich wprowadzenie: "Niepoprawnie wprowadzone dane. Proszę spróbować ponownie."

Wartości liczby kolumn i wierszy, punktu startowego, punktu końcowego oraz liczby podziału grafu należy podać jako liczby całkowite. W przeciwnym wypadku program zaokrągli je do najbliższej mniejszej liczby całkowitej będącej większą niż 0 (w przypadku podania liczby od 0 do 1 (bez 0) program domyślnie ustawia 1). Przy podaniu minimalnej wartości wag krawędzi większej niż wartość maksymalna, program wyświetli błąd oraz poprosi użytkownika o ponowne wprowadzenie danych: "Minimalna wartość wagi krawędzi jest większa od podanej maksymalnej wartości. Proszę wprowadzić dane ponownie."

Ustalenie punktu początkowego odpowiadającego punktowi końcowemu, skutkować będzie określeniem wartości najkrótszej drogi jako 0. Przy stworzeniu grafu i niesprecyzowaniu początkowego oraz końcowego węzła, program obierze punkt początkowy jako węzeł 1, a punkt końcowy jako węzeł $(x \times y)$.

Program umożliwia stworzenie grafu o wymiarach 1×1 . W takiej sytuacji początek i koniec grafu będą znajdowały się na tym samym miejscu o wartości 1. Wartość najkrótszej drogi między początkiem a końcem wyniesie 0. Program wypisze błąd po otrzymaniu argumentu n większego niż $\frac{x \times y}{4}$: "Niepoprawnie wprowadzona liczba podziału grafu (n)."

Przy losowym wyborze wierzchołka rozpoczynającego dzielenie możliwe jest uzyskanie maksymalnie podzielonego grafu w mniej niż $\frac{x \times y}{4}$ iteracjach. Program w tym wypadku wypisze w ilu iteracjach graf został maksymalnie podzielony. Program wypisze błąd gdy będziemy chcieli podzielić graf bez wcześniej podanych parametrów x oraz y . Program wypisze błąd podczas klikania przycisków "in", "Szukaj" oraz "Sprawdź spójność grafu" bez wcześniej podanych parametrów.

Wszystkie inne argumenty podane przez użytkownika, które przekraczają zdefiniowane wcześniej zakresy, będą skutkować wypisaniem błędu: "Wprowadzone dane przekraczają określone zakresy. Proszę spróbować ponownie."

Wszystkie powyższe komunikaty błędowe wyświetlane będą w polu tekstowym umieszczonym na samym dole okna programu.