

Projekt 26

Algorytm sprawdzający czy graf dwudzielny jest zupełny

Dokumentacja

Spis treści:

1. Opis działania aplikacji
 - 1.1. Ogólny opis
 - 1.2. Opis zawartości kodu
 - 1.3. Opis algorytmu
2. Instrukcja obsługi
 - 2.1. Menu główne
 - 2.2. Tworzenie nowego grafu
 - 2.3. Sprawdzanie czy graf jest zupełny

1. Opis działania aplikacji

1.1. Ogólny opis

Aplikacja pobiera plik zawierający dane grafu dwudzielnego (rozszerzenie *.graph*), sprawdza jego zupełność i zwraca wynik. Możliwe jest również stworzenie nowego grafu i zapisanie go jako plik *.graph*. Program umożliwia stworzenie tylko grafu dwudzielnego.

1.2. Opis zawartości kodu

W programie zawarte są następujące klasy:

- Vertex – klasa określająca wierzchołek. Zawiera jego numer oraz listę innych wierzchołków (*connectedWith*), z którymi jest połączony.
- Graph -klasa określająca graf. Zawiera listę wierzchołków (*AllVertecs*), które do niego należą.
- BipartiteGraph – klasa określająca graf dwudzielny. Zawiera dwa grafy: prawy (*Right*) i lewy (*Left*).

1.3. Opis algorytmu

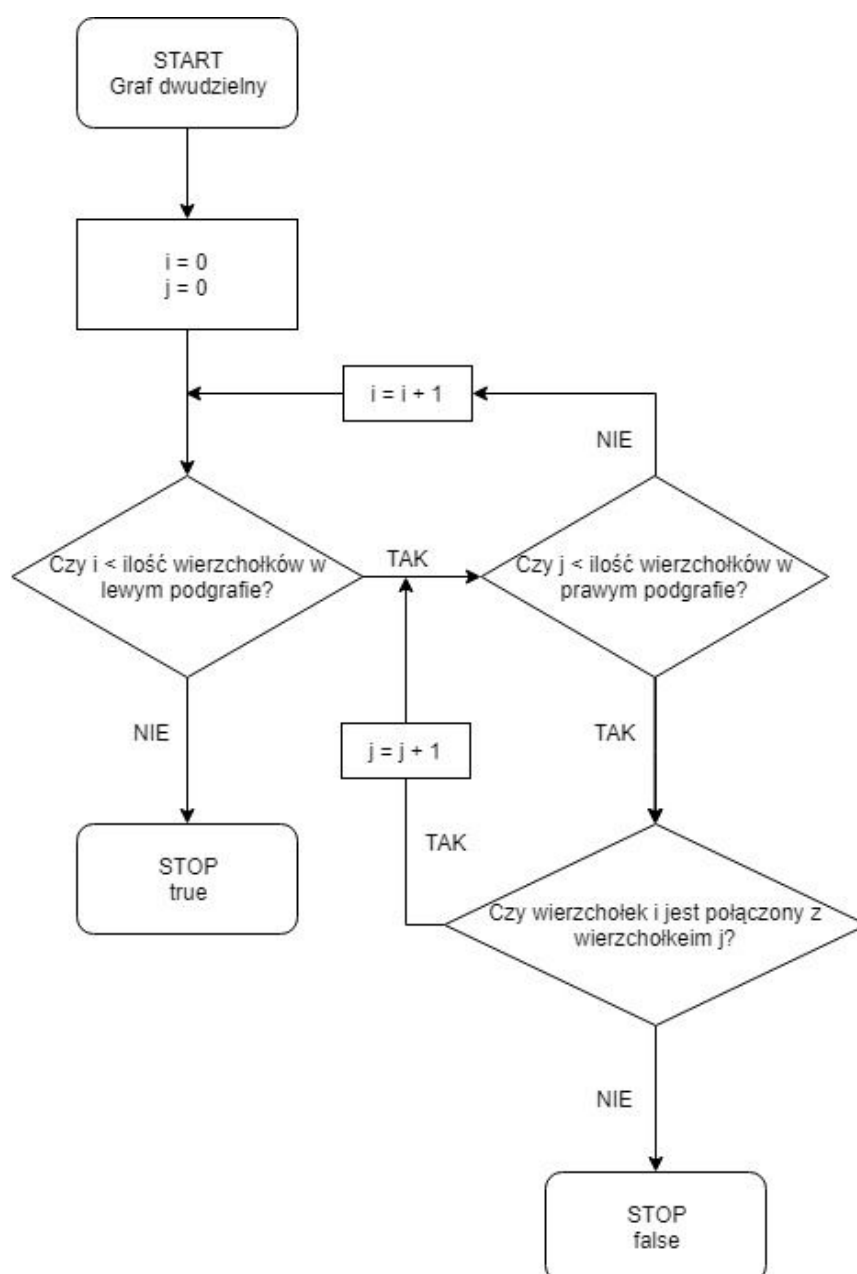
Algorytm *IsComplete* sprawdza czy podany graf dwudzielny jest zupełny. Na wejściu pobiera on dwudzielny graf a na wyjściu zwraca wartość typu *bool* (*true* – jeśli graf jest zupełny, *false* – jeśli nie jest). Algorytm *IsComplete* przedstawiono poniżej:

```
public static bool IsComplete(BipartiteGraph bipartiteGraph)
{
    for (int i = 0; i < bipartiteGraph.Left.AllVertecs.Count; i++)
    {
        for (int j = 0; j < bipartiteGraph.Right.AllVertecs.Count; j++)
        {
            if (!bipartiteGraph.Left.AllVertecs[i].connectedWith.Contains(bipartiteGraph.Right.AllVertecs[j]))
            {
                return false;
            }
        }
    }
    return true;
}
```

Oraz w postaci pseudokodu:

```
for każdy wierzchołek i należący do grafu lewego do
    for każdy wierzchołek j należący do grafu prawego do
        if wierzchołek i nie jest połączony z wierzchołkiem j
            then zwróć false
        fi
    od
od
zwróć true
```

Dla każdego wierzchołka należącego do lewego podgrafu w grafie dwudzielnym algorytm sprawdza czy ma on połączenie z każdym z wierzchołków podgrafu prawego. Jeśli nie zostanie odnalezione jakieś połączenie, algorytm zwróci *false*. Jeśli zaś zostaną odnalezione wszystkie połączenia, algorytm zwróci *true*. Poniżej przedstawiono schemat blokowy tego algorytmu.



2. Instrukcja obsługi.

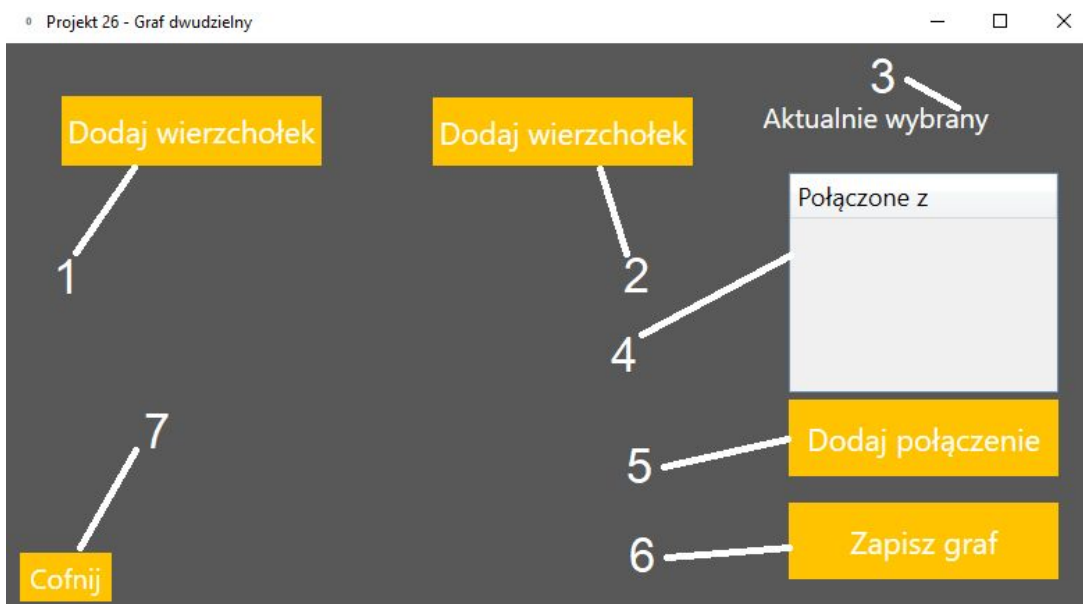
2.1. Menu główne.



Przycisk oznaczony jako (1) pozwala przejść do okna umożliwiającego stworzenie nowego grafu dwudzielnego. Utworzenie grafu jest konieczne, by móc sprawdzić jego zupełność.

Przycisk oznaczony jako (2) pozwala na przejście do okna sprawdzającego zupełność grafu. W oknie tym konieczne jest załadowanie wcześniej utworzonego grafu.

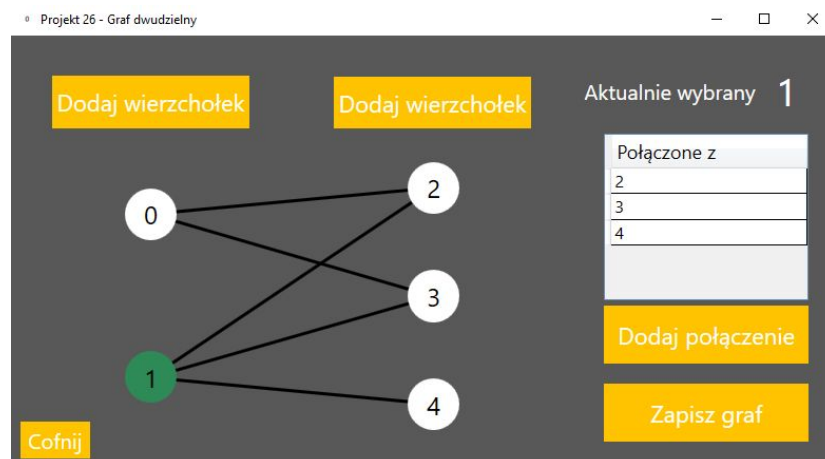
2.2. Tworzenie nowego grafu.



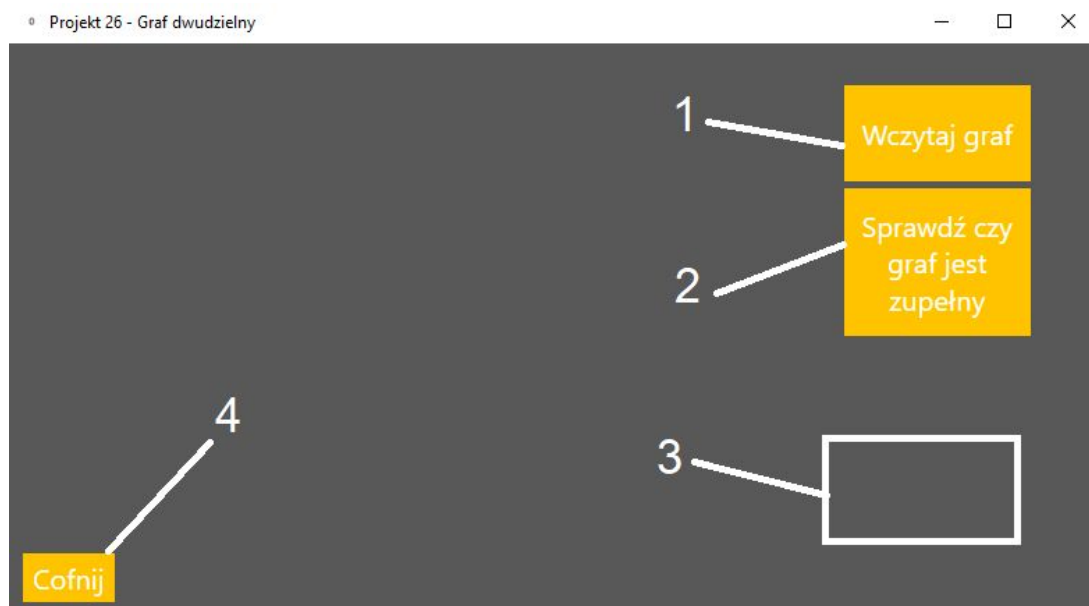
Przycisk oznaczony jako (1) pozwala na dodanie nowego wierzchołka do lewego podgrafu.

Przycisk oznaczony jako (2) pozwala na dodanie nowego wierzchołka do prawego podgrafu.

W polu oznaczonym jako (3) znajduje się numer aktualnie wybranego wierzchołka. Wierzchołek może zostać wybrany po kliknięciu na niego. W polu oznaczonym jako (4) znajduje się lista wierzchołków, z którymi aktualnie wybrany wierzchołek jest połączony. Przycisk oznaczony jako (5) pozwala dodać nowe połączenie do aktualnie wybranego wierzchołka. Po wciśnięciu przycisku podświetlony zostanie podgraf, z wierzchołkami którego można dodać połączenie. Przycisk oznaczony jako (6) pozwala na zapisanie utworzonego grafu, czyli wyeksportowanie go do nowo utworzonego pliku *.graph* o nazwie podanej przez użytkownika. Przycisk oznaczony jako (7) pozwala wrócić do menu głównego. Poniżej zamieszczono wizualizację tworzenia grafu:



2.3. Sprawdzanie czy graf jest zupełny.



Przycisk oznaczony jako (1) pozwala na wczytanie wcześniej stworzonego grafu (plik *.graph*) i jego wizualizację na ekranie. Przycisk oznaczony jako (2) uruchamia algorytm, który sprawdza czy wczytany graf jest zupełny, a następnie podaje odpowiedź. Pole oznaczone jako (3) jest miejscem, w którym algorytm zwróci odpowiedź.

Przycisk oznaczony jako (4) pozwala na powrót do menu głównego.
Poniżej przedstawiono wizualizację sprawdzania grafu:

