**Wydział Zastosowań Informatyki i Matematyki,   
SGGW, maj 2018**

Informatyka, semestr 4  
Grafy i sieci

**Projekt 13**  
Zuzanna Gałecka  
Bartosz Matyjasiak  
Maciej Zwoliński

**Zadanie:**

W oparciu o metodę poszukiwania maksymalnego przepływu w sieci opracować i zaimplementować algorytm najszybszego przepływu gazu w sieci pomiędzy jej dwoma punktami.

**Opis problemu maksymalnego przepływu:**

Problem maksymalnego przepływu to zagadnienie często spotykane w informatyce. Polega ono na znalezieniu dla danej sieci przepływowej takiego przepływu **f**, którego wartość jest maksymalna, gdzie wartość przepływu jest zdefiniowana jako łączny przepływ opuszczający źródło.   
Bardziej formalnie, dla danego przepływu **f**, w sieci **G = ( V , E )**, o źródle **s** i ujściu **t**, jego wartość jest zdefiniowana następująco:

Istnieje też uogólnienie tego problemu, w którym sieć **G** zawiera wiele źródeł i ujść.  
Innymi słowy , dodajemy do sieci **G** wierzchołek **s** połączony krawędziami o nieskończonej przepustowości ze wszystkimi źródłami oraz wierzchołek **t** połączony krawędziami o nieskończonej przepustowości ze wszystkimi ujściami. Wierzchołek **s** zwany jest **superźródłem**, zaś wierzchołek **t** - **superujściem**.

Jednym z algorytmów pozwalających rozwiązać problem maksymalnego przepływu jest algorytm Edmondsa-Karpa, na podstawie którego powstał nasz program.

Algorytm Edmondsa-Karpa jest jedną z realizacji metody Forda-Fulkersona rozwiązywania problemu maksymalnego przepływu w sieci przepływowej. Jego złożoność czasowa wynosi O(VE2), jest zatem wolniejszy od innych znanych algorytmów przepływowych działających w czasie O(VE3), takich jak algorytm relabel-to-front, czy algorytm trzech Hindusów. W praktyce jednak złożoność pesymistyczna rzadko jest osiągana, co w połączeniu z prostotą czyni algorytm Edmondsa-Karpa bardzo użytecznym, szczególnie dla grafów rzadkich.

Algorytm ten został odkryty przez rosyjskiego naukowca, E. A. Dinica w roku 1970, i niezależnie przez Jacka Edmondsa i Richarda Karpa w roku 1972. Artykuł Dinica zawiera dodatkowe techniki, które obniżają czas działania do O(V2E), (algorytm z tą poprawką nazywa się obecnie algorytmem Dynica).

Idea algorytmu jest identyczna z ideą metody Forda-Fulkersona, z dodatkowym warunkiem: ścieżka powiększająca, którą szukamy w każdym kroku algorytmu, musi być najkrótsza, czyli zawierać minimalną możliwą liczbę (nie wagę!) krawędzi. Taką ścieżkę znajduje się uruchamiając algorytm przeszukiwania grafu wszerz w sieci residualnej.

Poprawność algorytmu wynika wprost z twierdzenia Forda-Fulkersona: po zakończeniu działania w grafie nie może być ścieżki powiększającej, przepływ jest więc maksymalny. Dowód opiera się on na fakcie, że długość ścieżki powiększającej nie może maleć, a utrzymywać się na tym samym poziomie może przez co najwyżej **O(E)** kroków algorytmu (czyli jest co najwyżej **O(VE)** kroków, jako że długość ścieżki nie przekroczy **V**).

[źródło: wikipedia]

**ALGORYTM**:

**Ogólnie**:

1. Tworzymy sieć residualną,
2. Wyznaczamy ścieżkę powiększającą w sieci residualnej [przeszukując ją wszerz](http://algorytmy.ency.pl/artykul/przeszukiwanie_wszerz),
3. Jeśli nie da się wyznaczyć żadnej ścieżki powiększającej, kończymy działanie algorytmu. W przeciwnym razie modyfikujemy przepływ w sieci przepływowej o wartość residualną ścieżki powiększającej.

**Wejście:**

|  |  |  |
| --- | --- | --- |
|  | – | liczba wierzchołków w grafie |
|  | – | macierz *n* x *n* przepustowości kanałów |
|  | – | numer wierzchołka będącego źródłem sieci |
|  | – | numer wierzchołka będącego ujściem sieci |

**Wyjście:**

- macierz *n* x *n* przepływów netto

- wartość maksymalnego przepływu sieciowego

**Kroki**:

- Umieszczenie numeru źródła w kolejce

- Pobranie numeru wierzchołka z kolejki

- BFS

- Wyznaczenie przepustowości rezydualnej kanału

- Zapamiętanie poprzednika na ścieżce

- Obliczenie przepustowości rezydualnej do kolejnego węzła

- Znalezienie ścieżki rozszerzającej

- Zwiększenie przepływu sieciowego

- Cofanie się po ścieżce do źródła

- W kierunku przeciwnym zmniejszamy przepływ, w zgodnym zwiększamy

**ZASTOSOWANIA**:

- Wyznaczenie największego przepływu

- Znalezienie najdłuższej/najkrótszej drogi

**ZŁOŻONOŚĆ**:

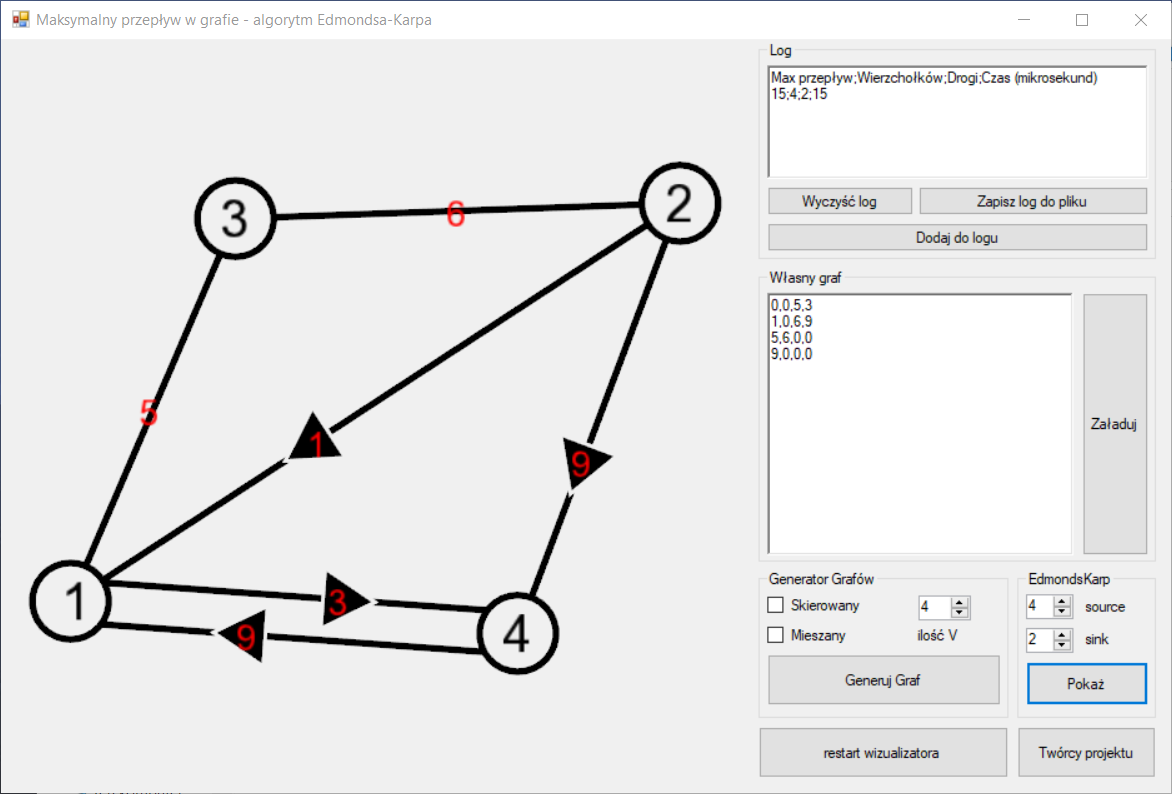
Wyznaczenie ścieżki za pomocą przeszukiwania wszerz ma złożoność *O(e)* (*e* jest liczbą krawędzi). Liczba wykonań głównej pętli algorytmu jest rzędu *O(ne)* (*n* jest liczbą wierzchołków). Złożoność czasowa algorytmu Edmondsa-Karpa wynosi zatem *O(ne2)*.

Wykonaliśmy testy złożoności czasowej naszego algorytmu dla kilku wygenerowanych automatycznie grafów. Wybraliśmy 7 różniących się od siebie liczba wierzchołków grafów i przedstawiliśmy różnice w złożoności czasowej oraz liczbie znalezionych dróg między nimi na wykresie:

Możemy zauważyć, że wykres czasu od l. wierzchołków przypomina parabolę o równaniu ax^2.

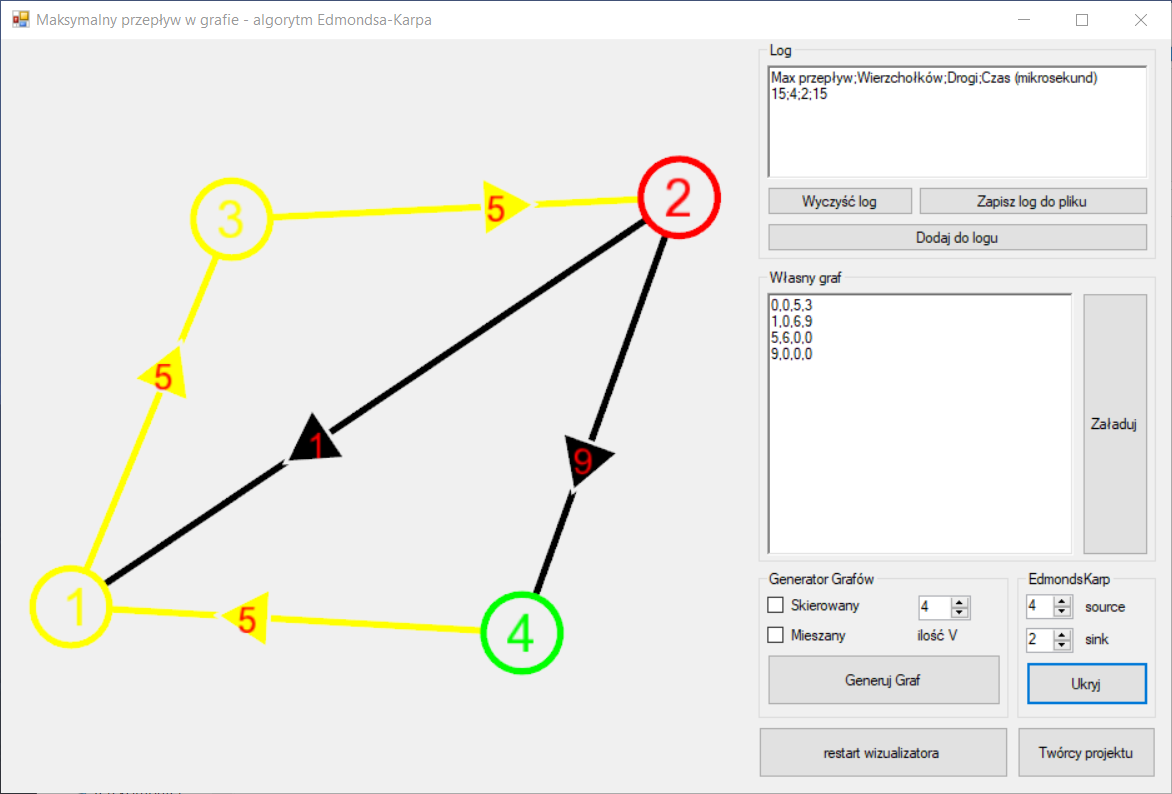
**Program okienkowy oraz konsolowy**

W celu wizualizacji grafu oraz działania algorytmu stworzono program okienkowy.

****

Program na następujące funkcje:

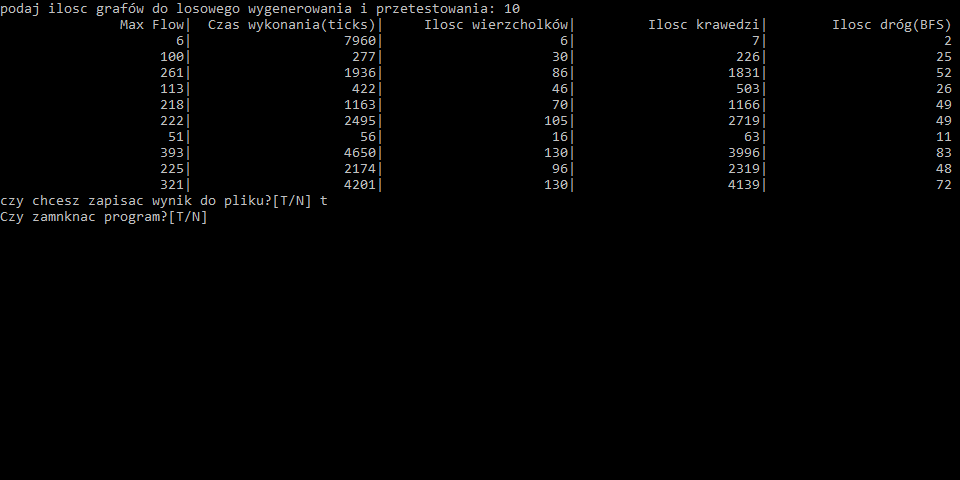
* Generowanie losowego grafu
  + Skierowanego (tylko z krawędziami skierowanymi)
  + Skierowanego (razem z krawędziami nieskierowanymi)
  + Nieskierowanego
* Rysowanie grafu zdefiniowanego przez użytkownika (macierz maksymalnych pojemności krawędzi)
* Pokazywanie wyniku działania algorytmu Edmondsa-Karpa z możliwością wyboru źródła oraz ujścia
* Dodawanie kolejnych wpisów do logu
  + Zapis logu do pliku
  + Wyczyszczenie logu

Jeśli użytkownik chce ujrzeć działanie algorytmu wystarczy, że wygeneruje graf lub wprowadzi swój własny, kliknie przycisk Pokaż w sekcji EdmondsKarp i wybierze źródło oraz ujście. Wtedy na ekranie powinien ukazać się graf z zaznaczonym źródłem na zielono, ujściem na czerwono oraz ścieżką przepływu na żółto.

Jeśli krawędź jest narysowana na żółto to liczba na niej odpowiada przepływowi przez tą krawędź, jeśli jest na czarno to liczba odpowiada maksymalnej pojemności przepływu na tej krawędzi.

By otrzymać informacje jaki jest maksymalny przepływ w grafie użytkownik musi dodać wpis do logu.

Do testów algorytmu stworzono program konsolowy



W tym programie możemy szybko przetestować algorytm dla dużej ilości losowych grafów, dla losowych źródeł i ujść, oraz zapisać wyniki do pliku.

W wynikach otrzymamy:

* Maksymalny przepływ (Max Flow)
* Czas wykonywania algorytmu w tick’ach systemu
* Ilość wierzchołków w grafie
* Ilość krawędzi
* Ilość dróg ze źródła do ujścia (pokazuje też, ile razy został wykonany algorytm BFS w algorytmie Edmondsa-Karpa)