**Wyszukiwanie najkrótszych połączeń kolejowych***Algorytm Floyda-Warshalla*

**Opis**

Celem projektu było utworzenie programu, który znajduje najkrótsze połączenia kolejowe pomiędzy wskazanymi miastami. Do osiągnięcia tego celu wykorzystano algorytm Floyda-Warshalla, którego podstawową wersję rozszerzono o zapamiętywanie przebiegu najkrótszej drogi pomiędzy dwoma dowolnymi wierzchołkami grafu.

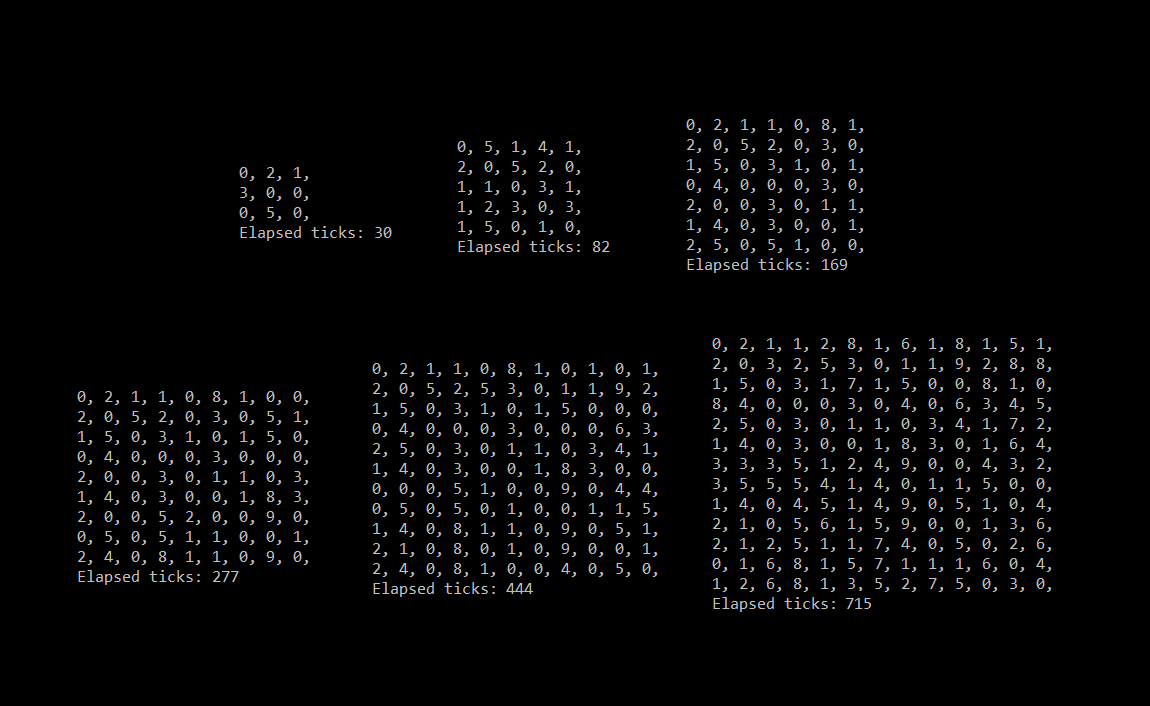
**Wydajność**

Zarówno złożoność czasowa i obliczeniowa algorytmu Floyda-Warshalla jest proporcjonalna do funkcji wielomianowej:

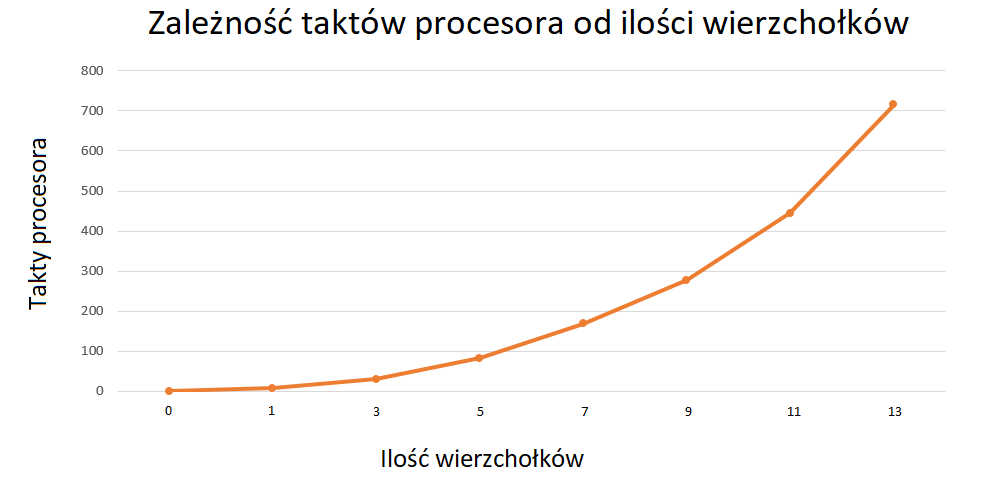
-złożoność czasowa: O(|V|3)  
-złożoność pamięciowa: O(|V|2)

gdzie |V| to liczba wierzchołków (liczba miast).

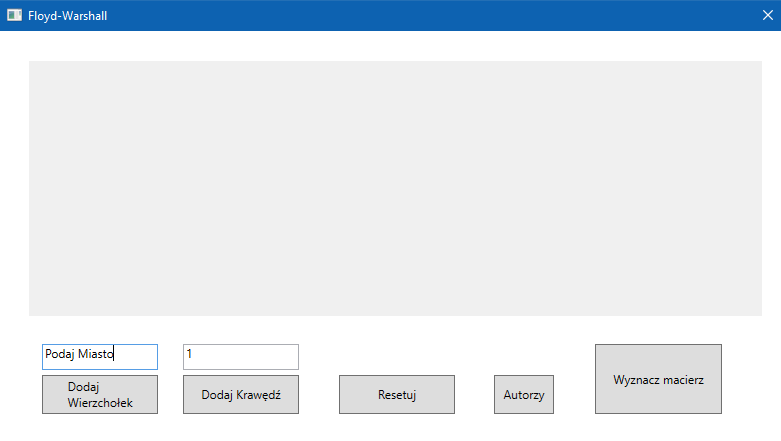
Przeprowadzone zostały testy wydajnościowe w aplikacji konsolowej



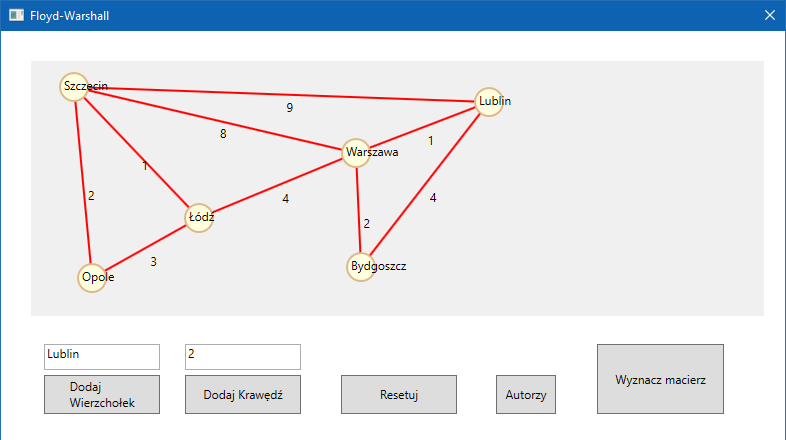
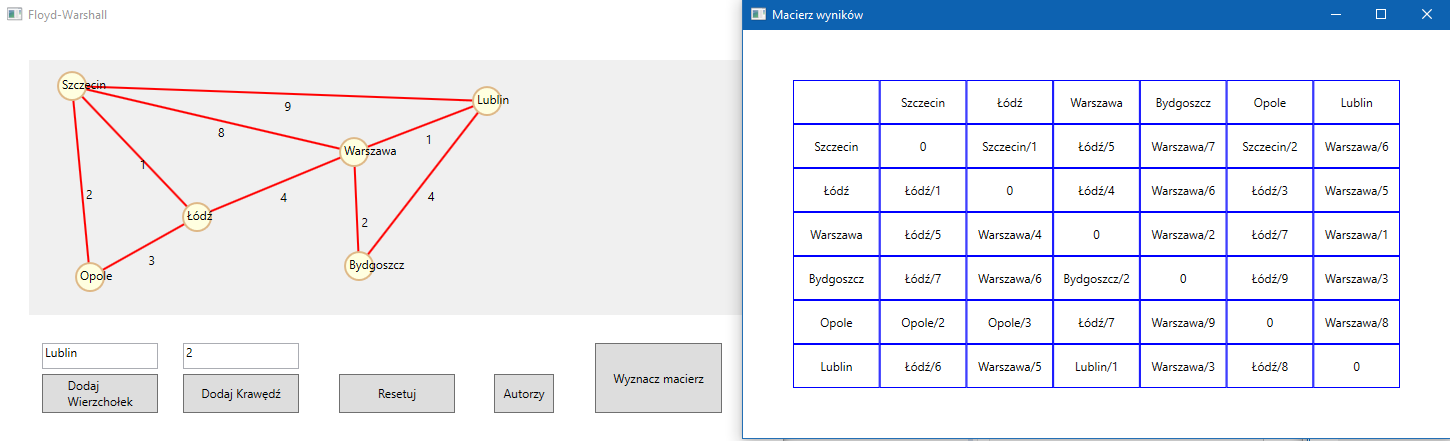
*Tiki procesora zmierzone podczas wykonywania algorytmu Floyda-Warshalla. Przedstawiono zależność ilości tików od rozmiaru przykładowych macierzy sąsiedztwa.*

* Zależność taktów procesora od ilości wierzchołków w grafie (ilości miast).*

**Instrukcja**



Program został napisany przy pomocy frameworka Microsoft .NET.  
Jest to prosta aplikacja pozwalająca wyznaczyć najkrótsze połączenia między kilkoma miastami.  
Użytkownik ma możliwość:

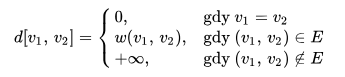
- podania nazwy miejscowości (jeśli tego nie zrobi zostanie dobrana losowa miejscowość) oraz umieszczenia tej miejscowości na obszarze rysowania  
- utworzenia połączenia miedzy dwoma miejscowościami oraz ustalenia kosztu dojścia między nimi (aby utworzyć połączenie należy kliknąć przycisk Dodaj Krawędź, a następnie lewym przyciskiem myszy kliknąć kolejno na element startowy i źródłowy)  
  
- przesuwania powstałych wierzchołków (należy trzymając lewy przycisk myszy przesunąć wybrany wierzchołek  
- usunięcia wszystkich wierzchołków i krawędzi poprzez kliknięcie przycisku Resetuj  
- zobaczenia, kto jest autorem projektu (przycisk Autorzy)  
- wyznaczenia macierzy wynikowej zawierającej nazwy miejscowości oraz najmniejsze koszty dojścia do nich oraz informację o poprzedniku (należy kliknąć przycisk Wyznacz macierz)  
  
Uwagi:

\* ze względu na temat projektu, uznaliśmy, że nie będziemy wykorzystywać grafów skierowanych  
\* dodatkowe zabezpieczenia nie pozwalają na wprowadzenie ujemnych wartości

**O algorytmie Floyda-Warshall’a** [źródło: wikipedia.org]

**Algorytm Floyda-Warshalla** wykorzystujący metodę [programowania dynamicznego](https://pl.wikipedia.org/wiki/Programowanie_dynamiczne) [algorytm](https://pl.wikipedia.org/wiki/Algorytm) służący do znajdowania najkrótszych ścieżek pomiędzy wszystkimi parami [wierzchołków](https://pl.wikipedia.org/wiki/Wierzcho%C5%82ek_(teoria_graf%C3%B3w)) w [grafie](https://pl.wikipedia.org/wiki/Graf_(matematyka)) ważonym. Graf może zawierać gałęzie zarówno o dodatniej i o ujemnej wadze („długości”) lecz nie może zawierać ujemnych cykli (cykli, w których suma wag krawędzi jest ujemna).

[Algorytm](https://pl.wikipedia.org/wiki/Algorytm) Floyda-Warshalla korzysta z tego, że jeśli najkrótsza ścieżka pomiędzy wierzchołkami *v1* i *v2* prowadzi przez wierzchołek *u*, to jest ona połączeniem najkrótszych ścieżek pomiędzy wierzchołkami *v1* i *u* oraz *u* i *v2*. Na początku działania algorytmu inicjowana jest tablica długości najkrótszych ścieżek, tak że dla każdej pary wierzchołków (*v1*,*v2*) ich odległość wynosi:

{\displaystyle d[v\_{1},\,v\_{2}]={\begin{cases}0,&{\mbox{gdy}}\ v\_{1}=v\_{2}\\w(v\_{1},\,v\_{2}),&{\mbox{gdy}}\ (v\_{1},\,v\_{2})\in E\\+\infty ,&{\mbox{gdy}}\ (v\_{1},\,v\_{2})\not \in E\end{cases}}}

Algorytm jest [dynamiczny](https://pl.wikipedia.org/wiki/Programowanie_dynamiczne) i w kolejnych krokach włącza do swoich obliczeń ścieżki przechodzące przez kolejne wierzchołki. Tak więc w *k*-tym kroku algorytm zajmie się sprawdzaniem dla każdej pary wierzchołków, czy nie da się skrócić (lub utworzyć) ścieżki pomiędzy nimi przechodzącej przez wierzchołek numer *k* (kolejność wierzchołków jest obojętna, ważne tylko, żeby nie zmieniała się w trakcie działania programu). Po wykonaniu |*V*| takich kroków długości najkrótszych ścieżek są już wyliczone.

**Zapis w pseudokodzie**

Dla grafu *G* i funkcji wagowej *w* otrzymamy tablicę *d*[*v1*][*v2*] odległości pomiędzy wierzchołkami *v1* i *v2*.

Floyd-Warshall(*G*,*w*)

**dla każdego** wierzchołka v1 w V[G] **wykonaj**

**dla każdego** wierzchołka v2 w V[G] **wykonaj**

d[v1][v2] = nieskończone

poprzednik[v1][v2] = niezdefiniowane

d[v1][v1] = 0

**dla każdej** krawędzi (v1,v2) w E[G]

d[v1][v2] = w(v1,v2)

poprzednik[v1][v2] = v1

**dla każdego** wierzchołka u w V[G] **wykonaj**

**dla każdego** wierzchołka v1 w V[G] **wykonaj**

**dla każdego** wierzchołka v2 w V[G] **wykonaj**

**jeżeli** d[v1][v2] > d[v1][u] + d[u][v2] **to**

d[v1][v2] = d[v1][u] + d[u][v2]

poprzednik[v1][v2] = poprzednik[u][v2]

**Projekt wykonali:**

- Michał Kocisz  
- Kamil Paździorek  
- Grzegorz Jarząbek