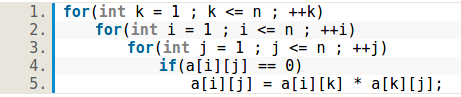
**Roy-Floyd**

Algoritmul Roy-Floyd este folosit in diverse domeni des intalnite, rolul principal fiind acela de gasire a drumului de cost minim intre un obiect principal si o tinta anume.

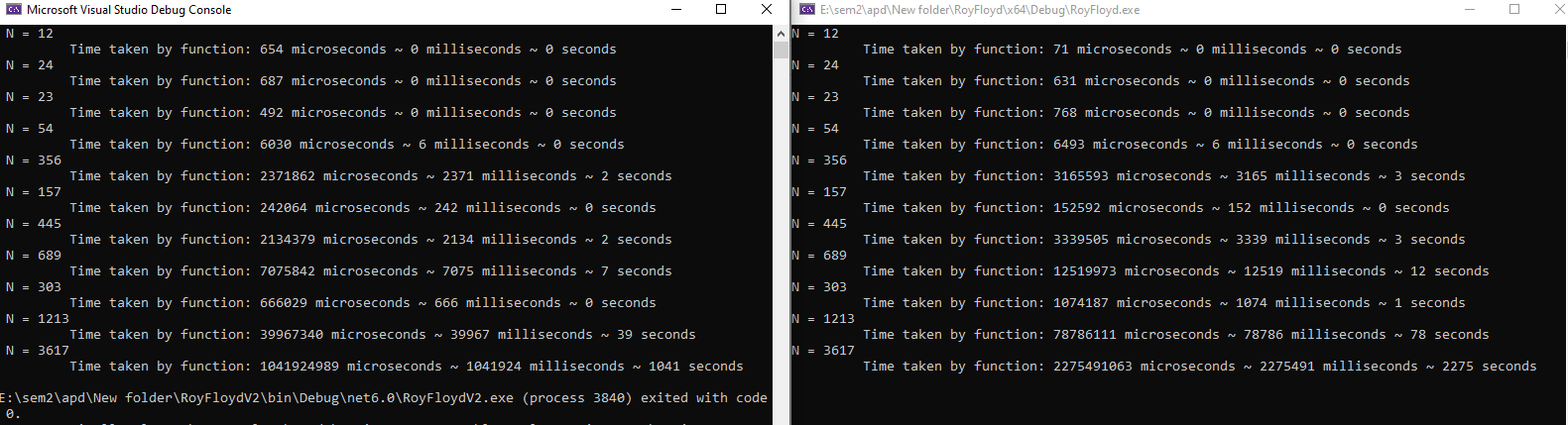
Algoritmul Floyd-Warshall(intalnit si sub numele de Roy-Floyd) compara toate drumurile posibile din graf dintre fiecare 2 noduri, si poate fi utilizat si in grafuri cu muchii de cost negativ.  
  
Complexitatea temporala este O(|V|^3), iar cea spațială este tot O(|V|^3). O complexitate spațială cu un ordin mai mic se obține observând ca la un pas nu este nevoie decât de matricea de la pasul precedent d[i, j, k-1] si cea de la pasul curent d[i, j, k]. O observație și mai bună este că, de la un pas k-1 la k, estimările lungimilor nu pot decât sa scadă, deci putem sa lucram pe o singura matrice. Deci, spațiul de memorie necesar este de dimensiune |V|^2.

Implementarea secventiala:



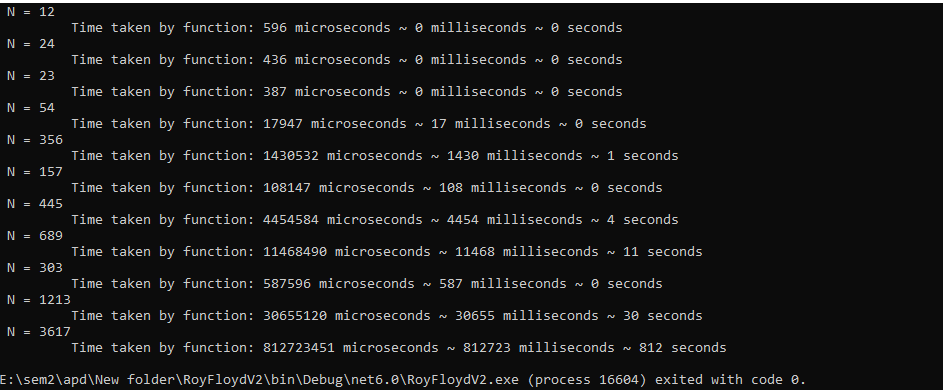
Am implementat acest algoritm in 2 limbaje:

* C++
* C# (console application cu .net 6)
* Python

Rezultatele au fost surprinzatoare. Rezultatele de la 2 rulari (C# si C++) le puteti vedea mai jos.   
  
  


In poza din stanga sunt rezultatele in urma rularii programului scris in C#, respective poza din dreapta contine rezultatele rularii proogramului scris in C++.  
Observam ca pentru date mari, C# este aproape de 2 ori mai rapid decat C++.   
Nu ma asteptam la aceste rezultate.

In graficele urmatoare putem vedea rezultatele executiilor algoritmului Roy-Floyd in cele 3 limbaje.

Iata alte rezultate si mai bune pentru date mari (C#)  
  


Rezultatele rularii algoritmului in Python

