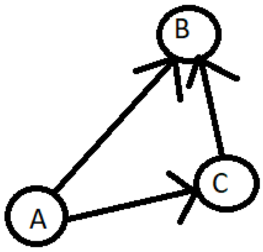
Algorytm Floyda - Warshalla do znajdowania najkrótszych ścieżek pomiędzy wierzchołkami grafu.

Cechy:

1) Trzeba przeanalizować od razu cały graf (tj. znaleźć najkrótsze ścieżki pomiędzy wszystkimi węzłami).
2) Powyższe wiąże się z dość dużą złożonością obliczeniową $O(V^3)$ oraz pamięciową $O(V^2)$, gdzie V to liczba wierzchołków w grafie.
3) W przeciwieństwie do algorytmu Dijkstry działa poprawnie także dla ujemnych wag krawędzi.
4) W grafie nie może być jednak cykli, dla których suma wag jest ujemna (bo wtedy algorytm działa źle)
5) Algorytm może jednak wykrywać takie cykle (tj. o ujemnej sumie krawędzi).

Podstawowe założenie:



Jeśli odległość pomiędzy węzłami AB (d_{AB}) jest większa niż suma odległości d_{AC} oraz d_{CB} to z A do B krótsza droga prowadzi przez węzeł C niż bezpośrednio od A do B.

Pseudokod:

1) Tworzymy dwuwymiarową (kwadratową) tablicę X o rozmiarze równym liczbie wierzchołków w grafie. Uwaga: stąd bierze się złożoność pamięciowa algorytmu $O(V^2)$.
2) Wypełniamy macierz X nieskończonościami za wyjątkiem diagonali, którą wypełniamy 0. Oznacza to tyle, że dla każdego wierzchołka jego odległość od samego siebie wynosi 0, a odległość od pozostałych jest nieskończenie duża.
3) Musimy przeiterować przez wszystkie krawędzie grafu. Dla każdej pojedynczej krawędzi (łączącej wierzchołek i-ty z j-ym) w macierzy X w pozycji X[i][j] wstawiamy wagę tej krawędzi.
4) Teraz mamy potrójnie zagnieżdżoną pętlę:
```
Dla każdego wierzchołka grafu k
        Dla każdego wierzchołka grafu i
                Dla każdego wierzchołka grafu j
                        Jeśli X[i][j]>X[i][k]+X[k][j]
                                To X[i][j]=X[i][k]+X[k][j]
```
5) Po wykonaniu tej pętli w tablicy X mamy najkrótsze odległości pomiędzy poszczególnymi wierzchołkami.
6) Aby sprawdzić, czy w grafie są cykle o ujemnej liczbie krawędzi trzeba przeanalizować elementy na diagonali macierzy. Jeśli dla dowolnego i, na diagonali leży element mniejszy od 0 to oznacza, że graf ma ujemny cykl.