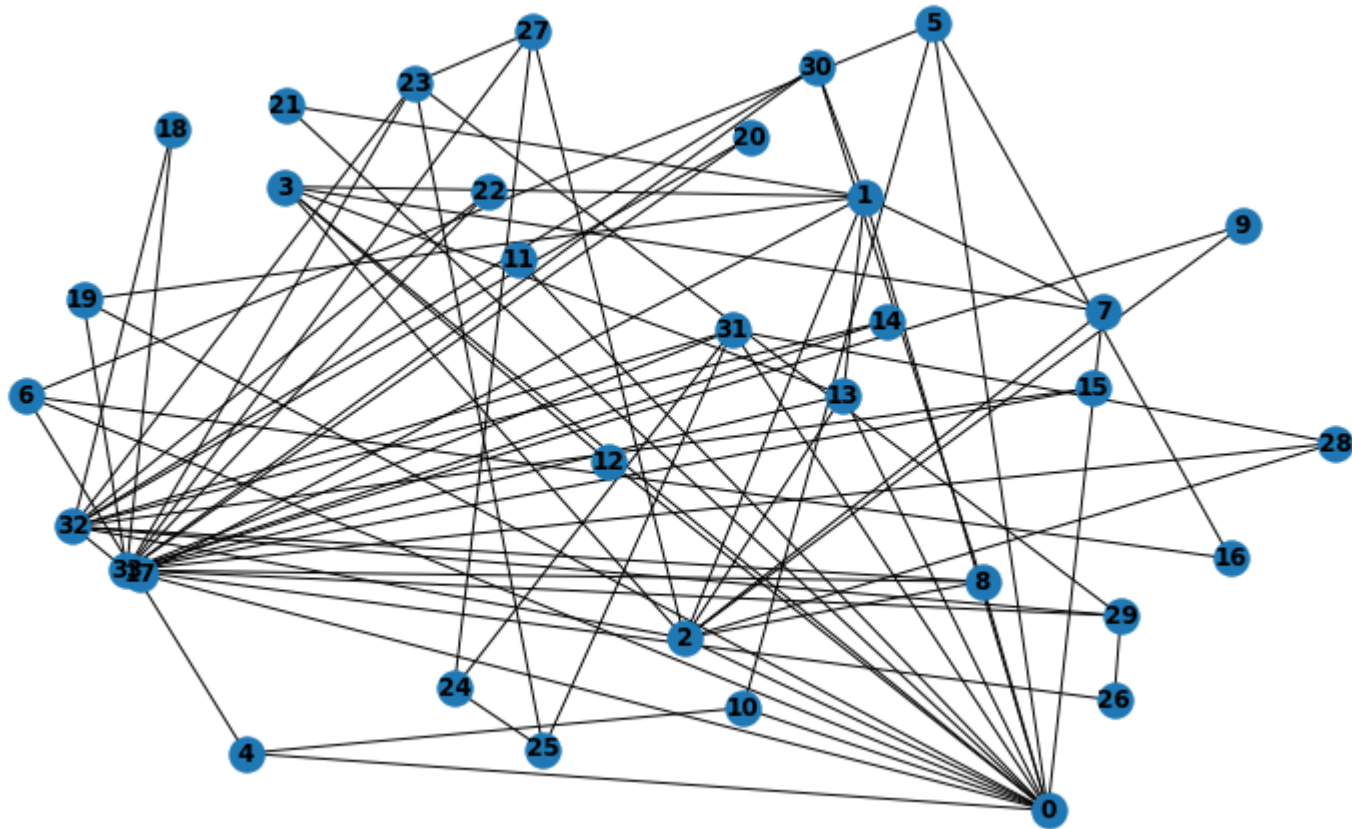


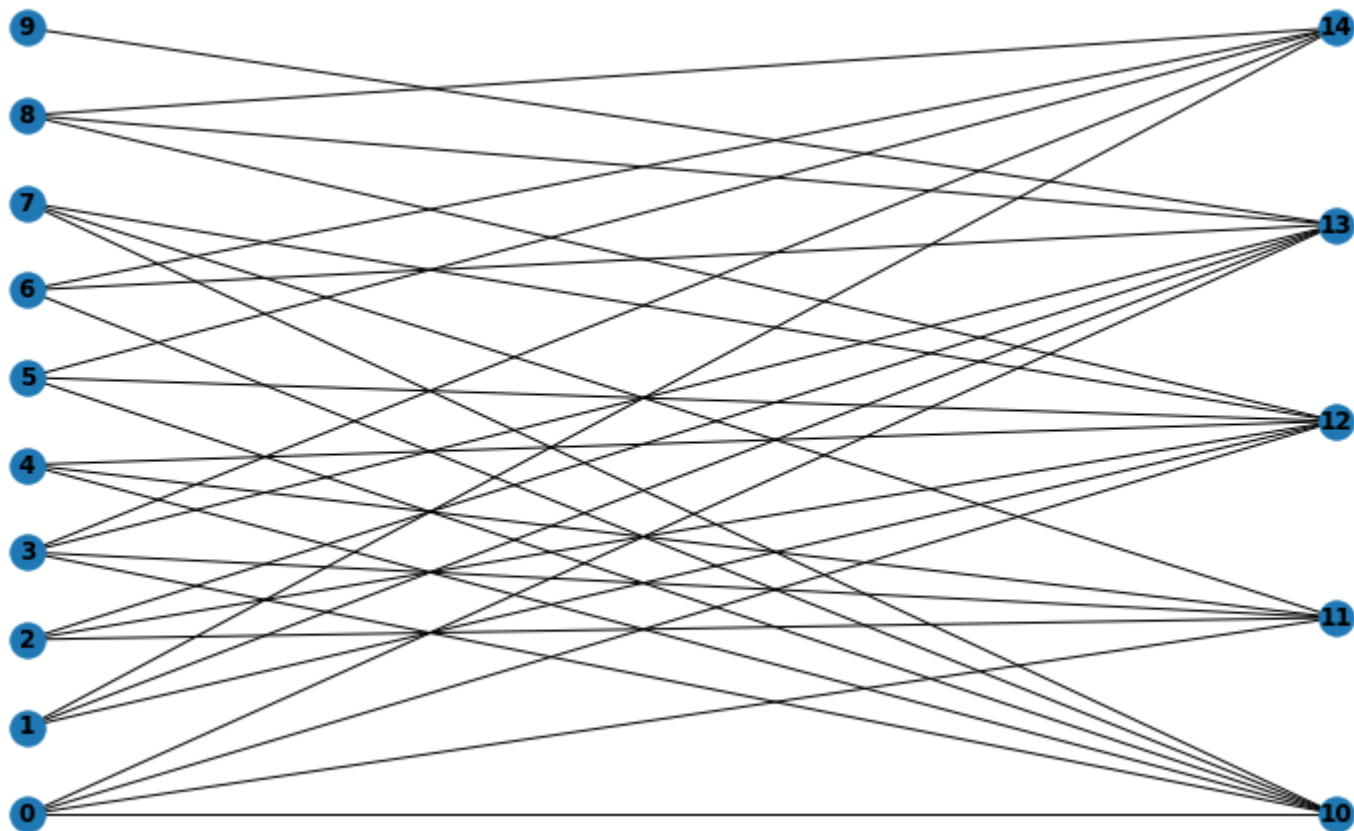
Algorytmy wizualizacji grafów

27.10.2020

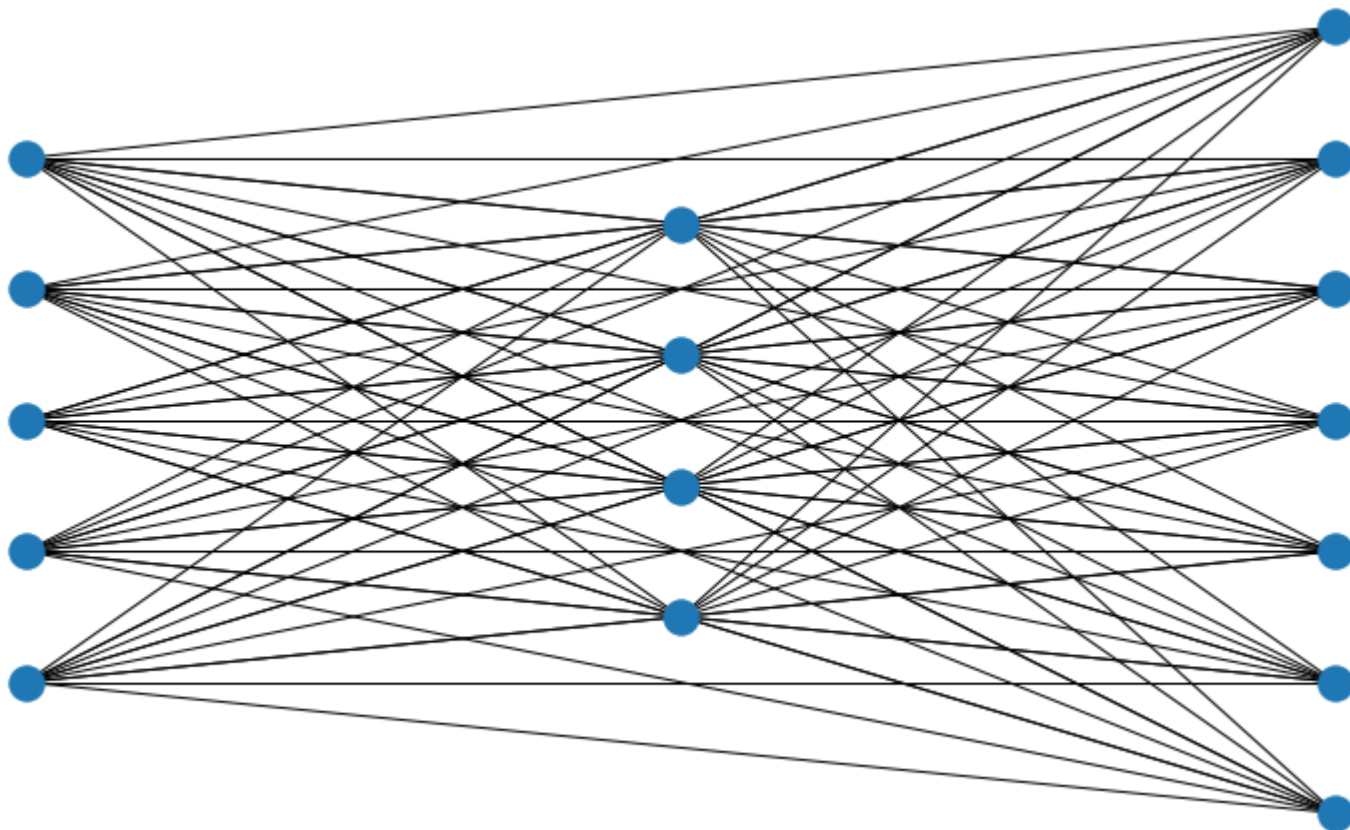
Random layout



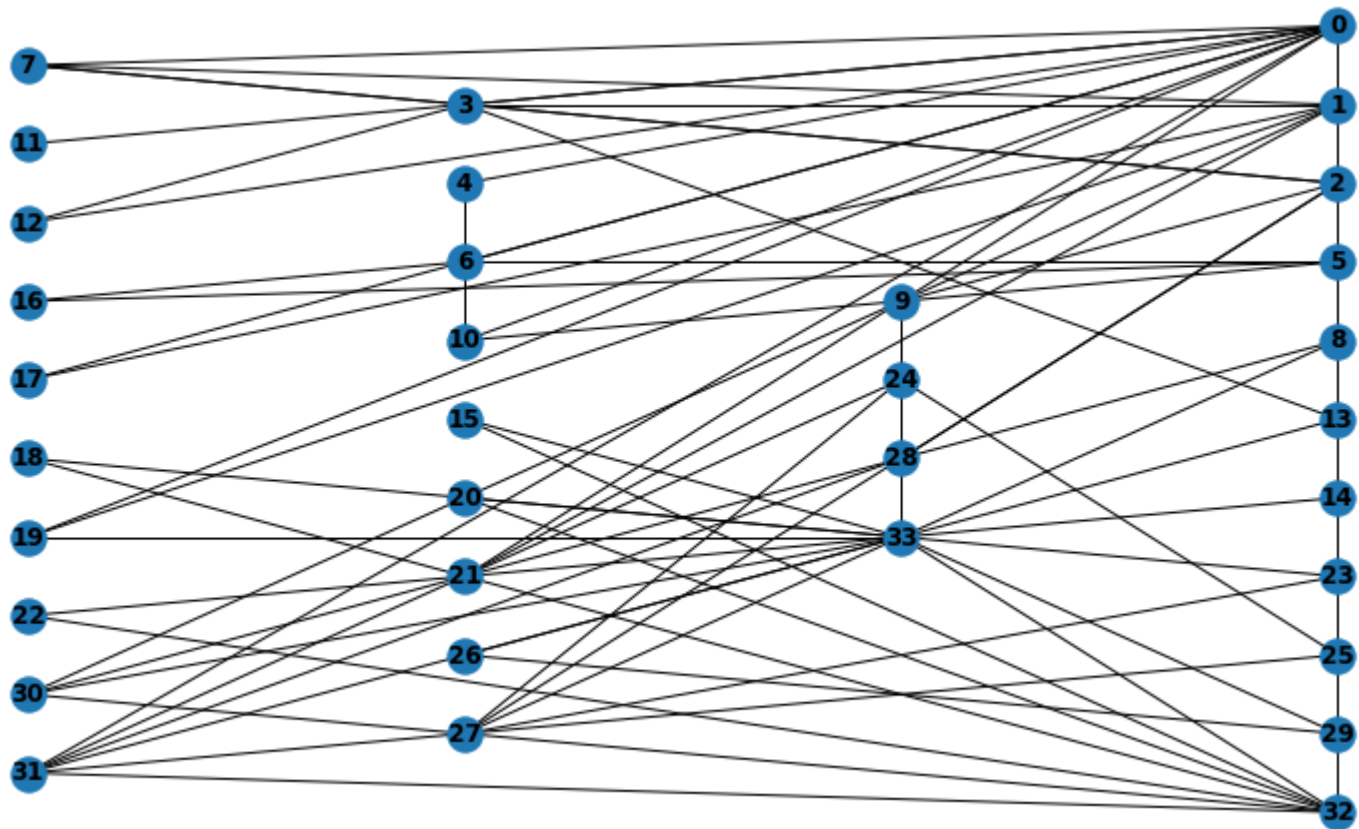
Bipartite layout



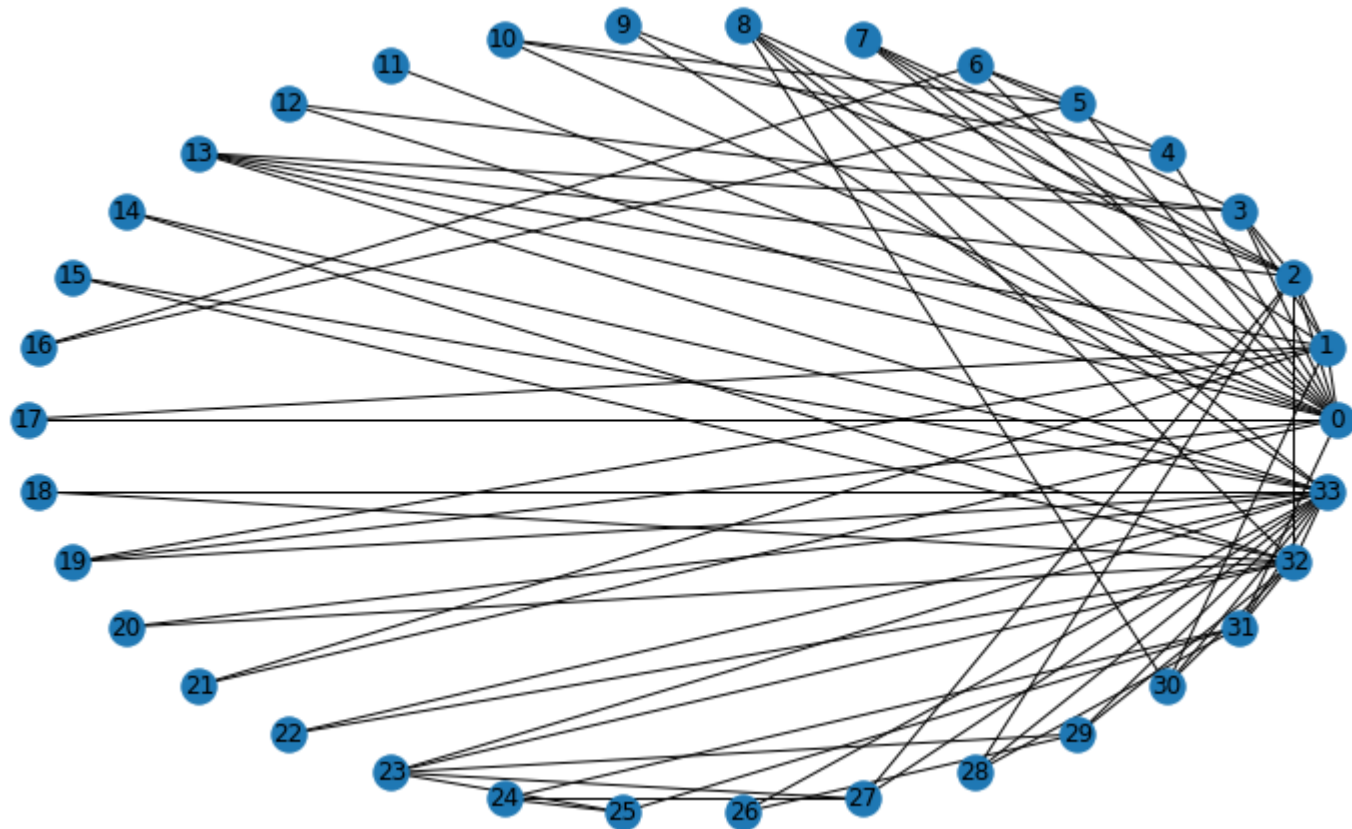
Multipartite layout



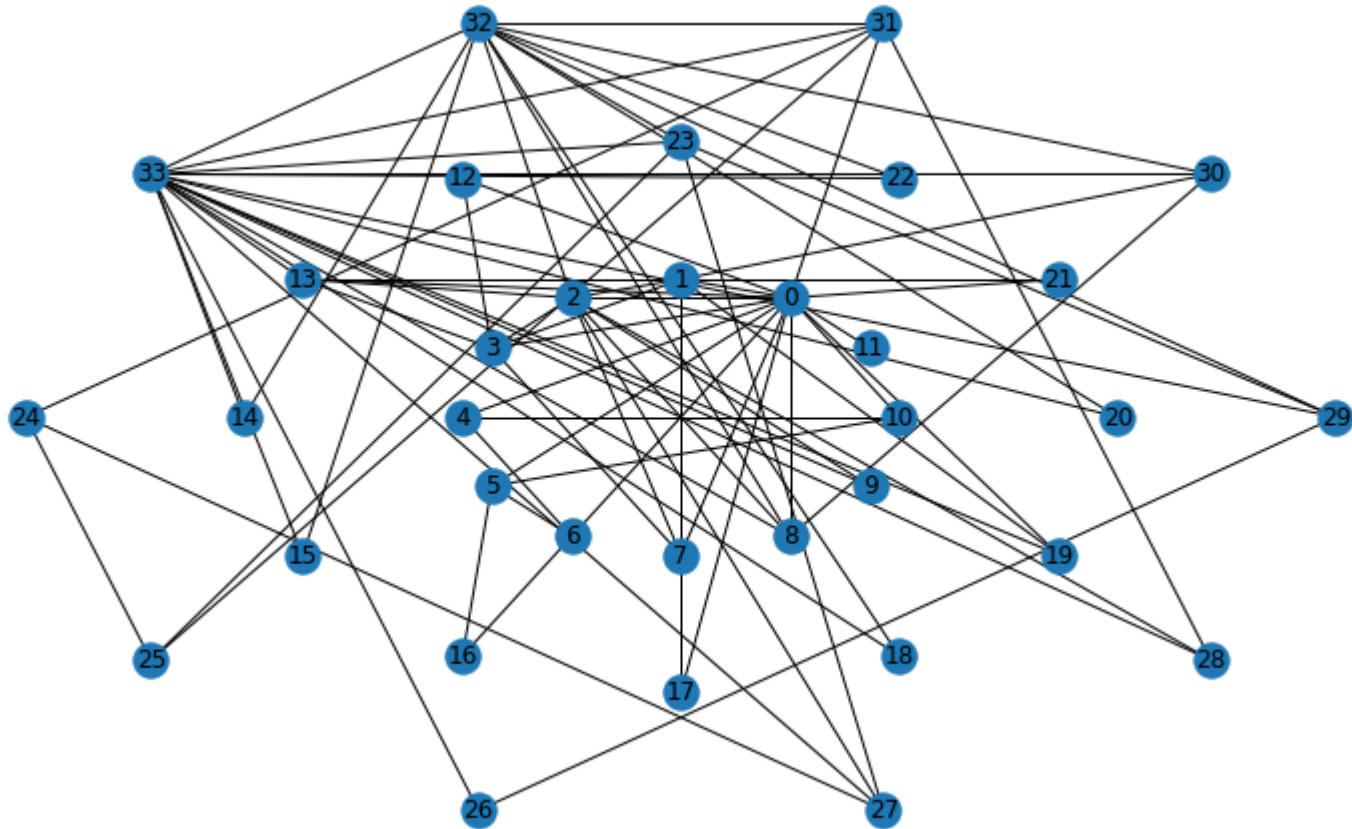
Multipartite layout



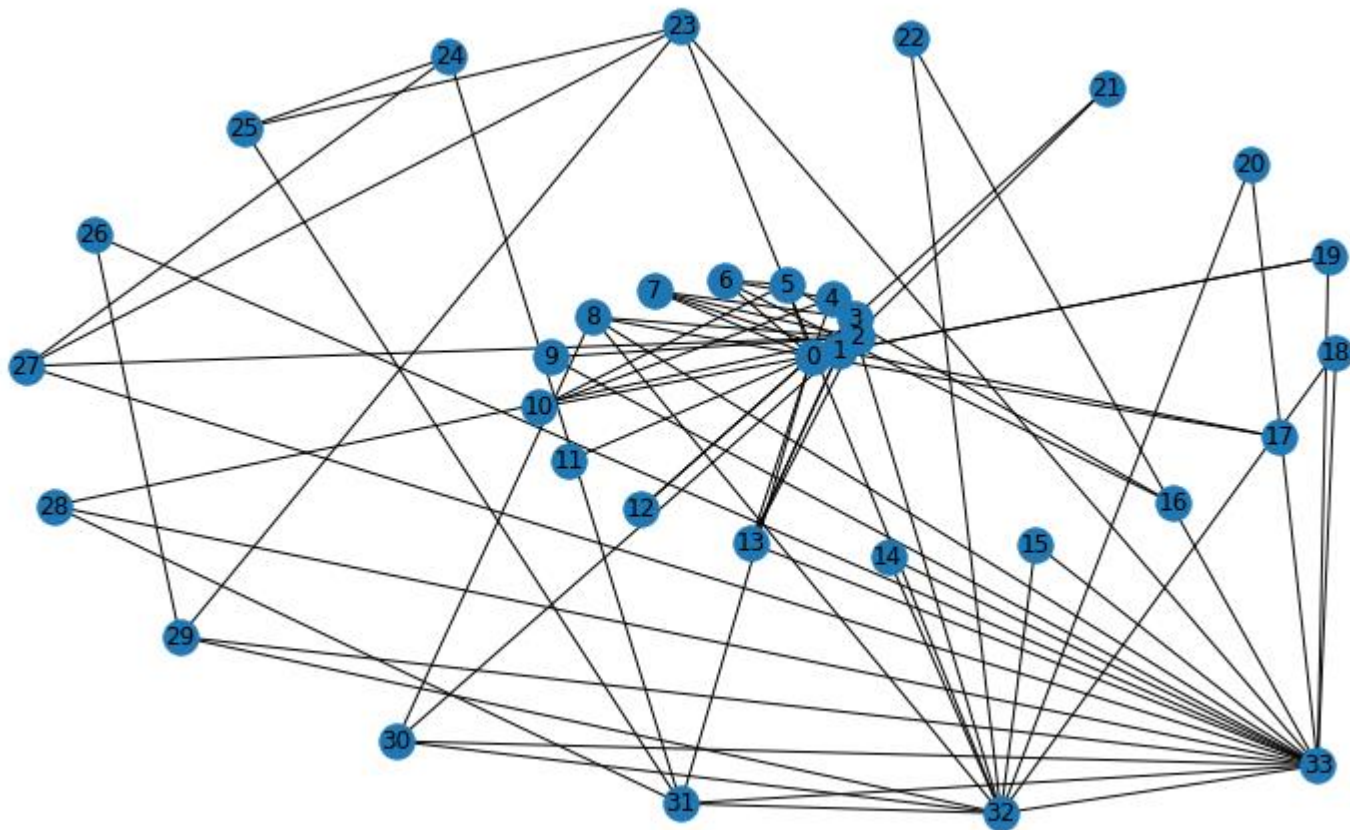
Circular layout



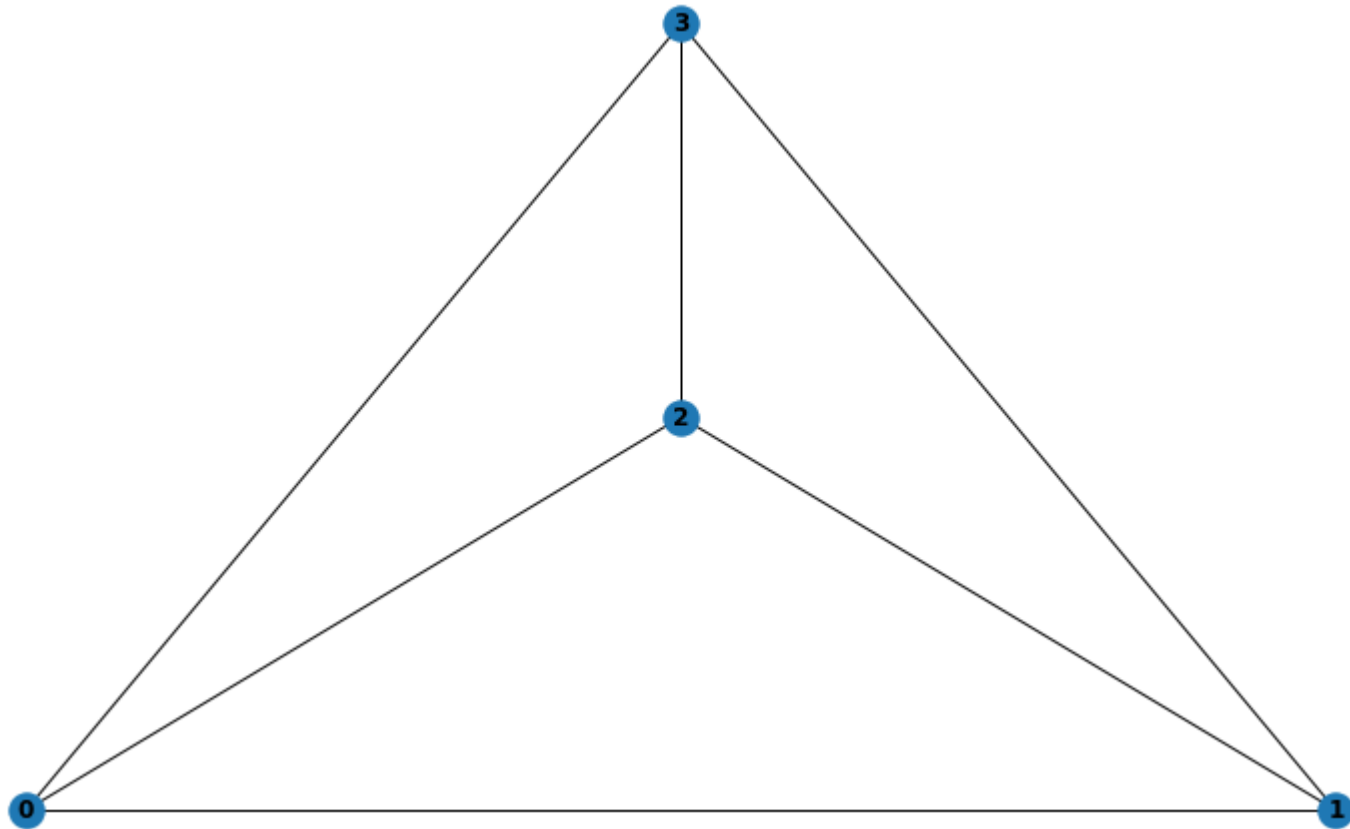
Shell layout



Spiral layout



Planar layout



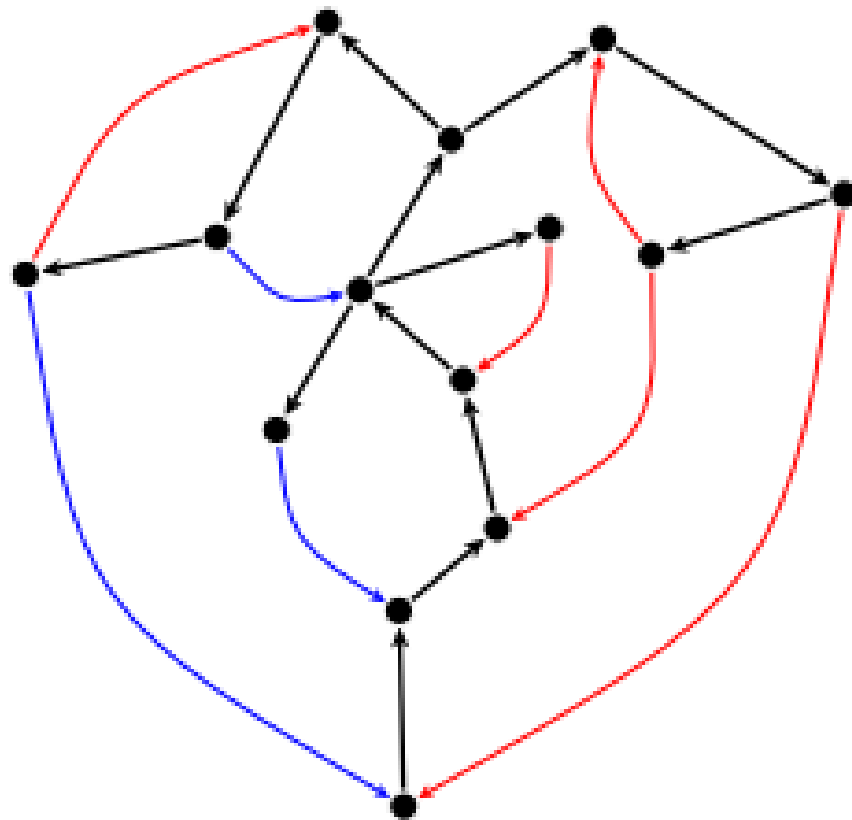
The left-right planarity test

- ☐ algorytm opiera się na obserwacji, że cykle są fragmentami grafu, które mogą spowodować przecinanie się krawędzi, w szczególności różne cykle mające wspólne krawędzie
- ☐ cykle dzielą płaszczyznę na 2 rozłączne części, należy zatem przemyśleć czy pozostałe wierzchołki w grafie łączymy “wewnątrz” czy “na zewnątrz” cyklu
- ☐ sprawdzenie planarności sprowadza się do podjęcia powyższej decyzji dla każdego cyklu, to jest określenie czy kierunek cyklu jest zgodny czy niezgodny ze wskazówkami zegara. Jeśli można to zrobić bez przecięcia cykli, graf jest planarny
- ☐ cykli w grafie może być bardzo dużo, jednak wystarczy skupić się tylko na reprezentatywnym podzbiorze cykli

The Left-Right Planarity Test

Reprezentatywny podzbiór cykli:

- ☐ weźmy skierowane drzewo rozpinające, które tworzymy jak w DFS
- ☐ określamy relatywny do drzewa kierunek krawędzi powracających:
 - lewy - krawędzie niebieskie,
 - prawy - krawędzie czerwone
- ☐ otrzymujemy reprezentatywny podzbiór cykli wraz z ich skrętnością

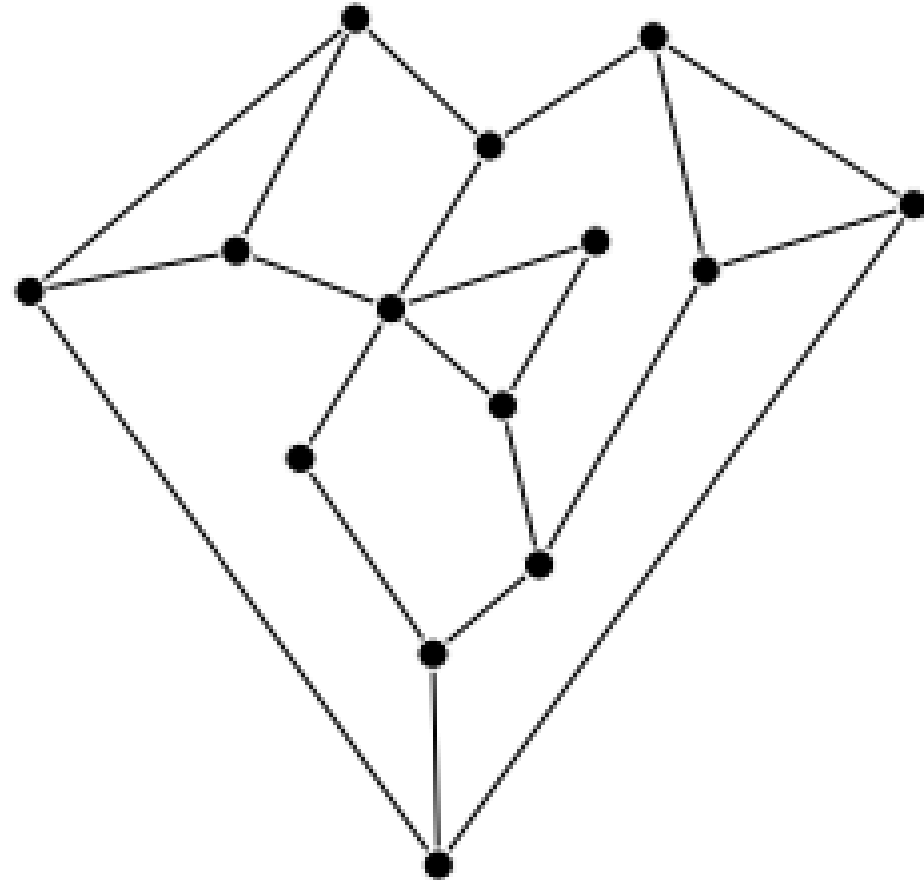
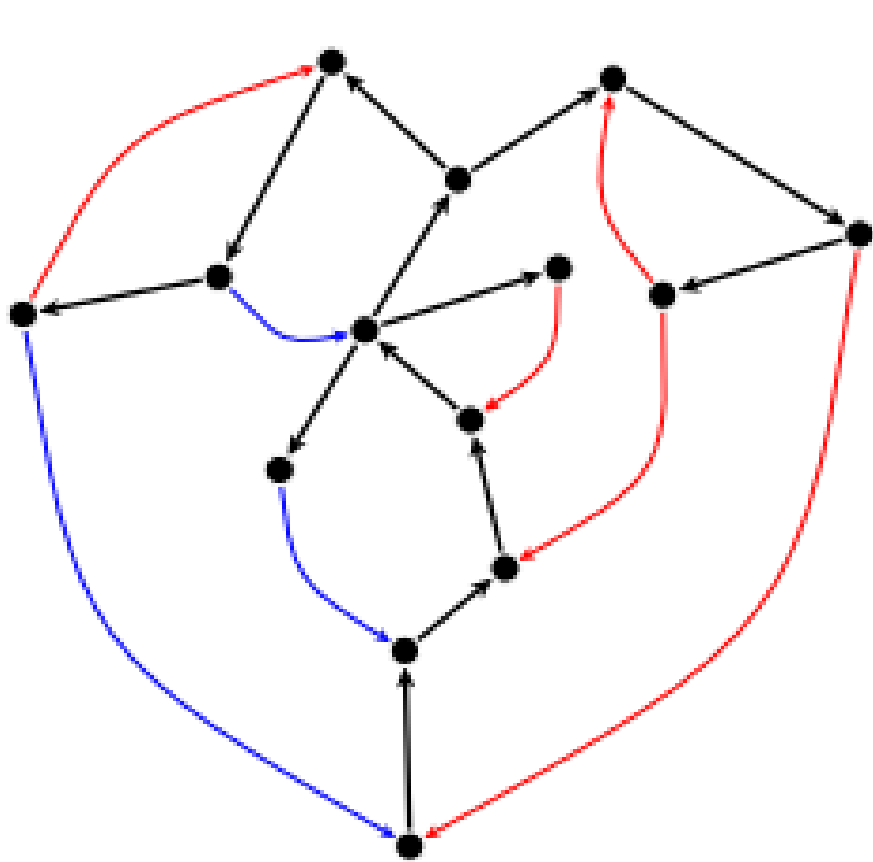


Papier wyjaśniający szczegóły:

Ulrik Brandes: The Left-Right Planarity Test 2009

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.217.9208>

The Left-Right Planarity Test

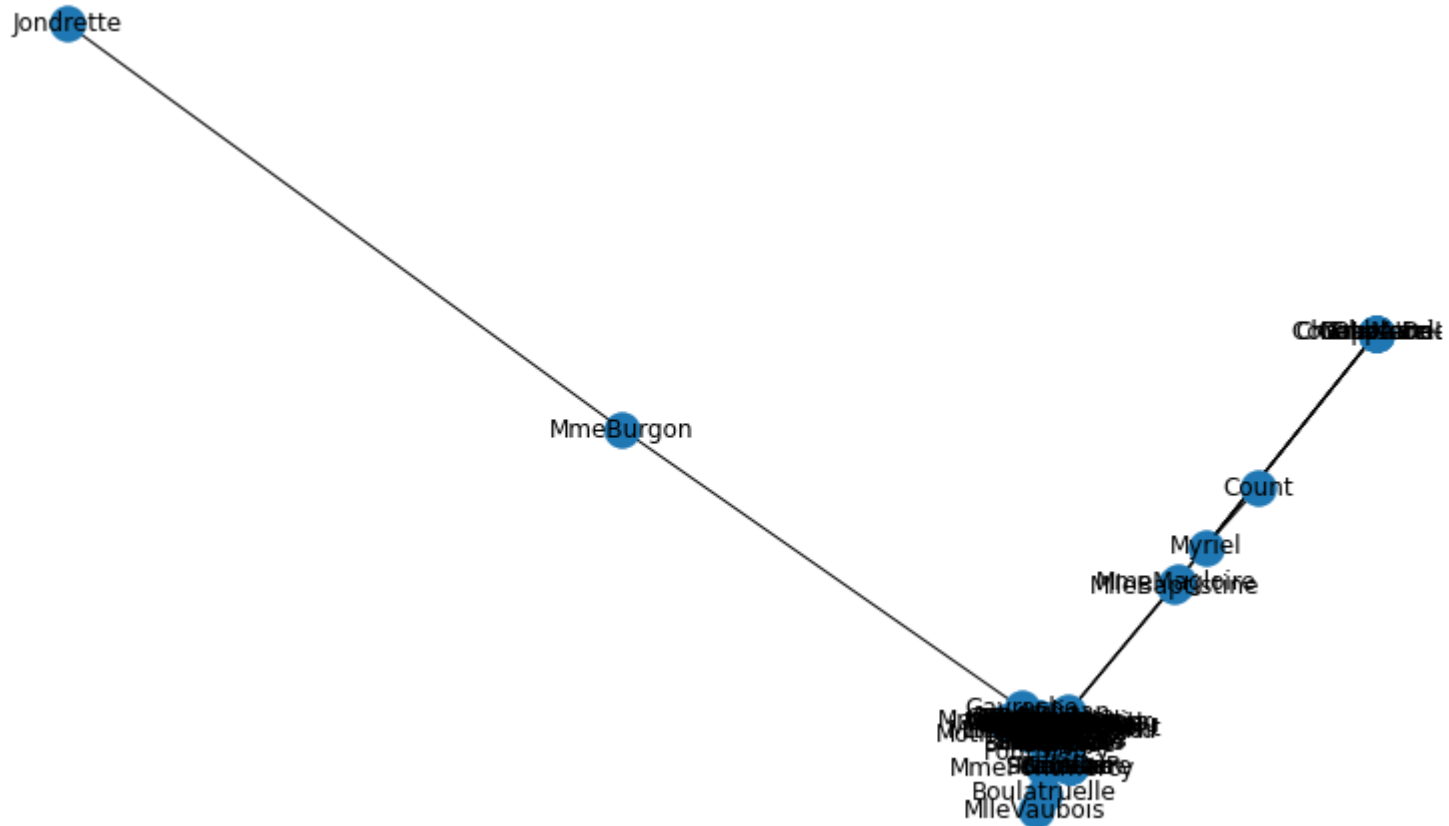


Papier wyjaśniający szczegóły:

Ulrik Brandes: The Left-Right Planarity Test 2009

<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.217.9208>

Spectral layout



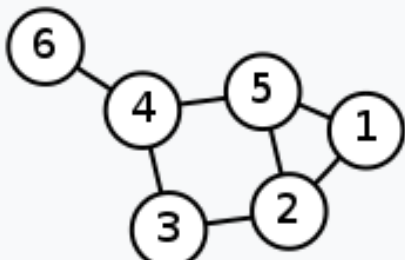
Spectral layout

Klastruje wierzchołki korzystając z wektorów własnych macierzy Laplace'a.

Macierz Laplace'a

$$L = D - A$$

gdzie D to macierz stopni wierzchołków, A to macierz sąsiedztwa grafu.

GRAF	D	A	L
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

Spectral layout

- policzmy 2 najmniejsze niezerowe wartości własne ($\lambda_2 \leq \lambda_3$) macierzy L

λ_2 - spectral gap, mierzy rozpiętość spektralną grafu

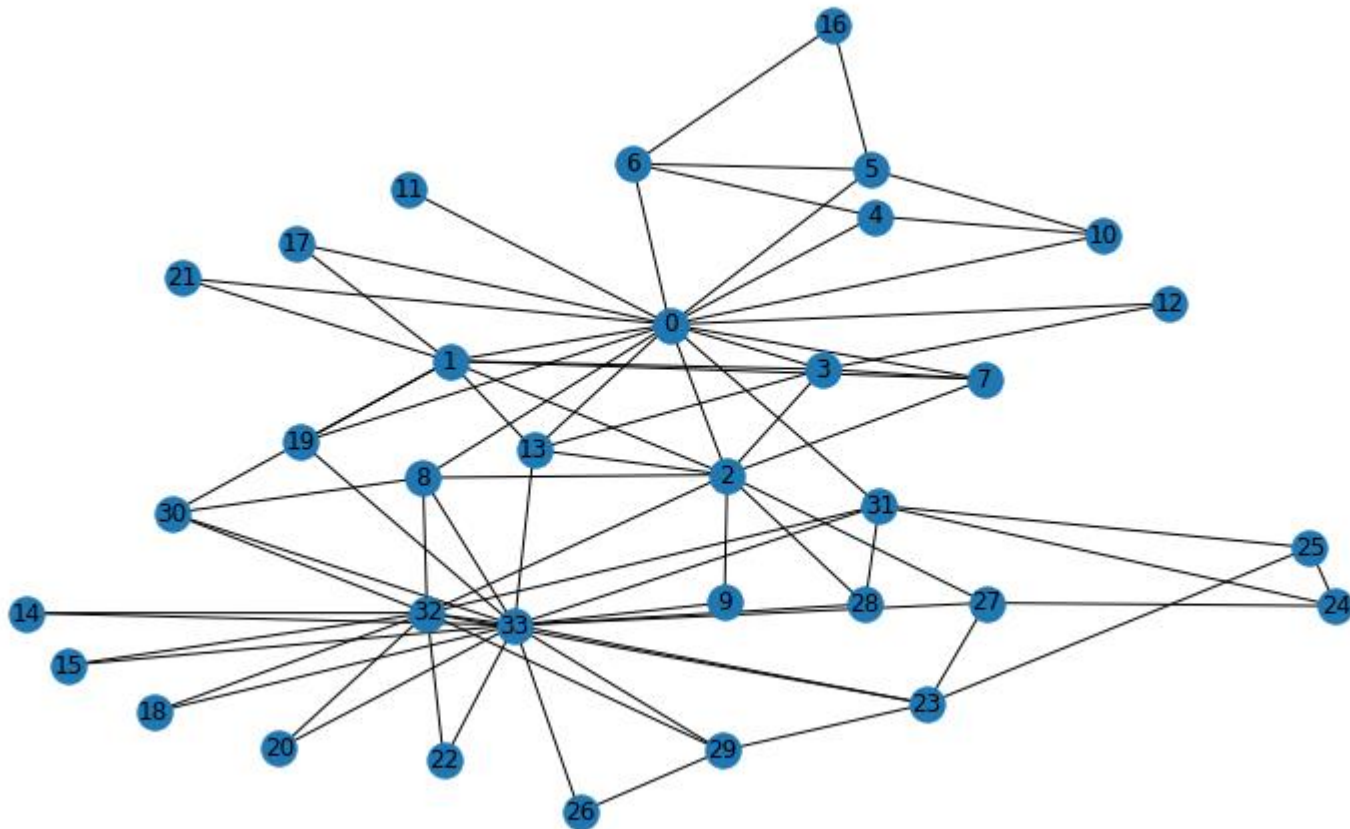
λ_3 - wartość Fiedler'a, miara zagęszczenia krawędzi w grafie

- policzmy znormalizowane wektory własne u_2, u_3 związane z odpowiednio λ_2 and λ_3

$$R = [u_2 \ u_3]$$

- wierzchołek i -ty umieszczamy na płaszczyźnie o koordynatach i -tego wiersza macierzy R

Kamada-Kawai layout



Kamada-Kawai layout

Pomysł:

- ☐ Każdy wierzchołek - cząsteczka połączona sprężynami ze wszystkimi innymi
- ☐ Długości sprężyn - proporcjonalne do najkrótszych ścieżek między wierzchołkami
- ☐ Szukamy równowagi

T. KAMADA and S. KAWAI "AN ALGORITHM FOR DRAWING GENERAL UNDIRECTED GRAPHS" (1989)
[http://www.itginsight.com/Files/paper/AN%20ALGORITHM%20FOR%20DRAWING%20GENERAL%20UNDIRECTED%20GRAPHS\(Kadama%20Kawai%20layout\).pdf](http://www.itginsight.com/Files/paper/AN%20ALGORITHM%20FOR%20DRAWING%20GENERAL%20UNDIRECTED%20GRAPHS(Kadama%20Kawai%20layout).pdf)

Kamada-Kawai layout

Minimalizuje

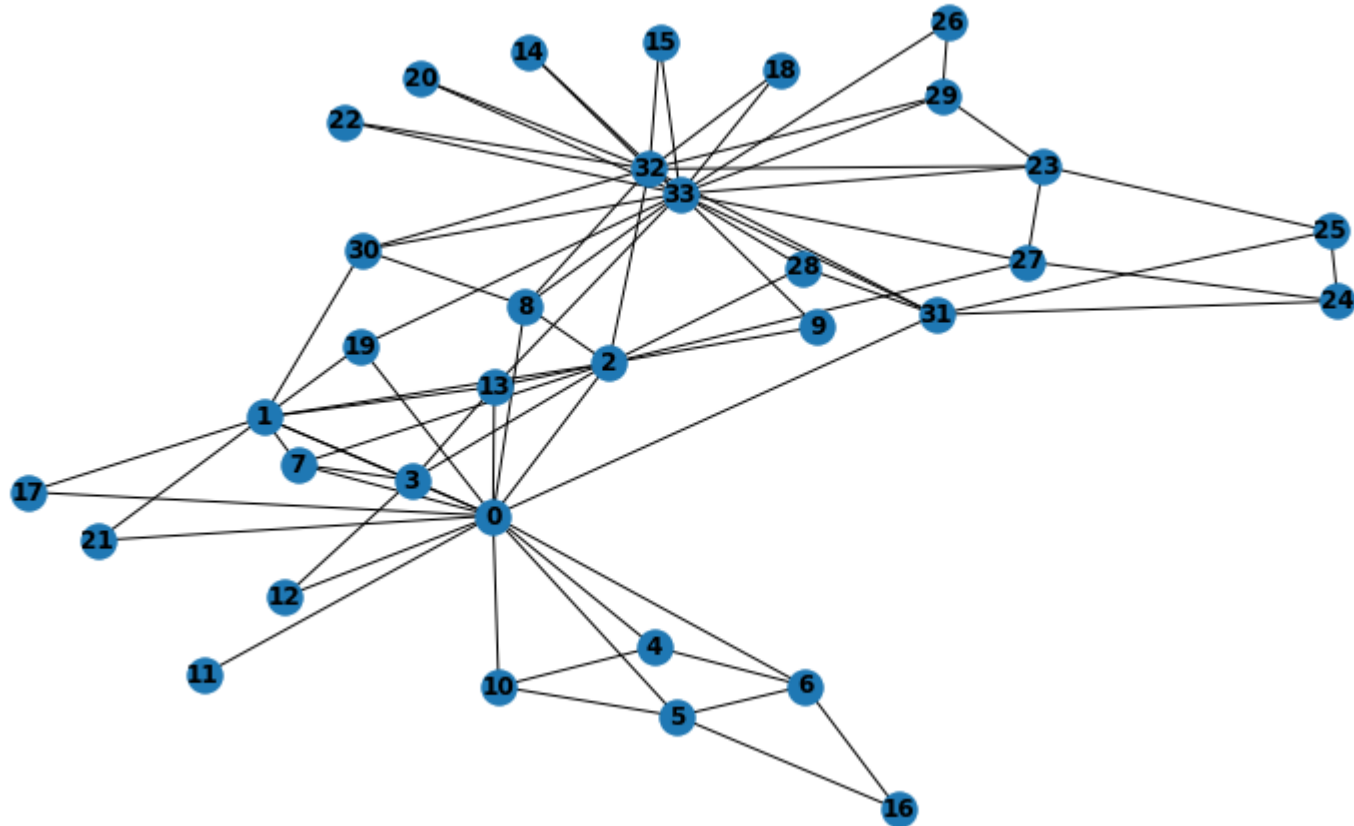
$$E = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \frac{1}{2} k_{ij} (|p_i - p_j| - l_{ij})^2.$$

$$l_{i,j} = d_{i,j} * L$$

$d_{i,j}$ - najkrótsza ścieżka między węzłami

L - oczekiwana długość wyświetlonej krawędzi

Spring Layout



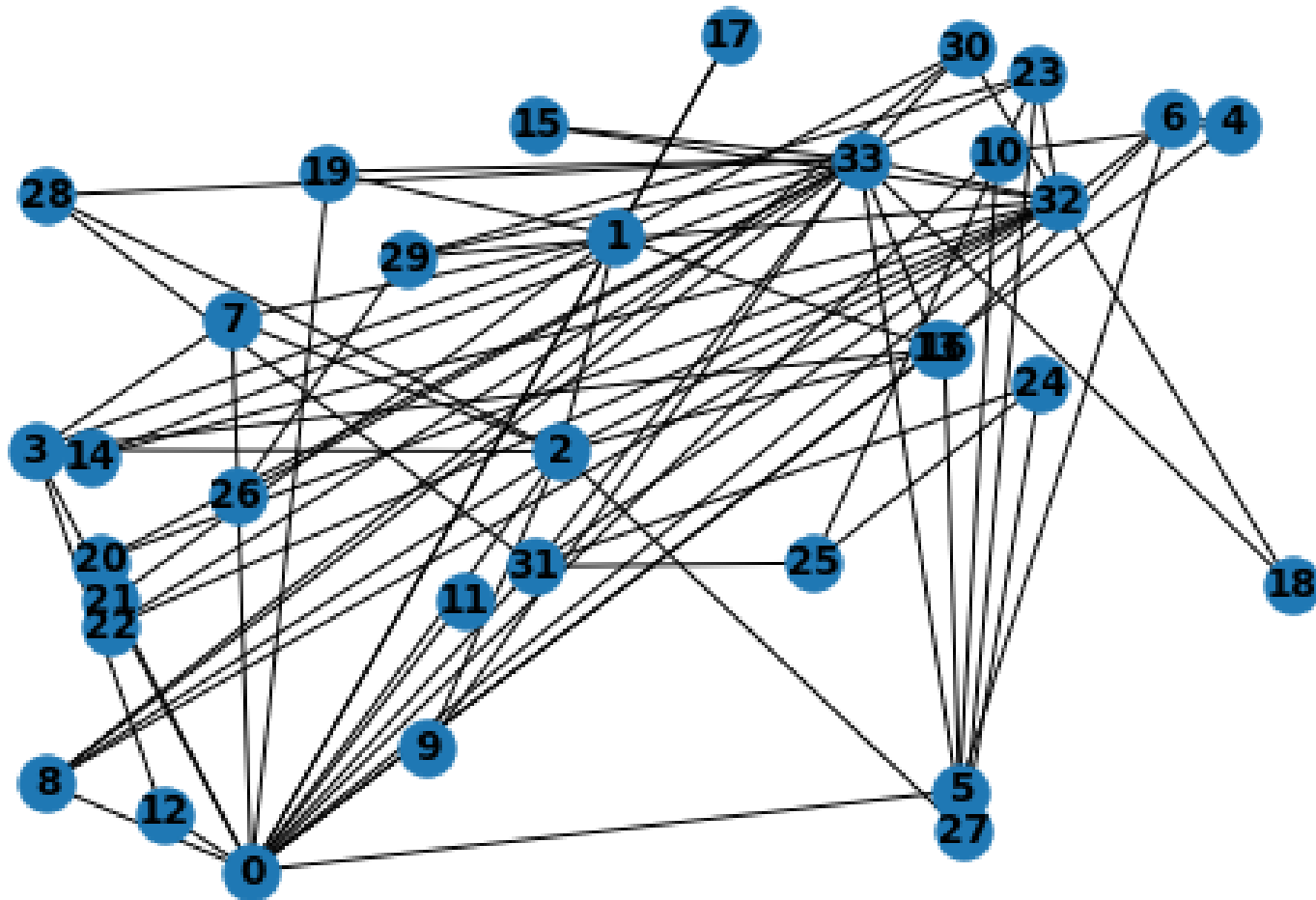
Spring Layout

Algorytm Fruchterman-Reingold:

- ☐ symuluje zachowanie modelu, w którym wierzchołki są stalowymi pierścieniami a krawędzie reprezentują sprężyny mocujące pierścienie do siebie
- ☐ siły przyciągające w modelu są analogiczne do działania sprężyn a siły odpychające do sił elektrostatycznych
- ☐ w każdej iteracji ruch wierzchołka zależy od wypadkowej działających na niego sił przyciągania i odpychania
- ☐ temperatura układu, to jest ruchliwość wierzchołków, maleje z każdą iteracją a układ zmierza ku equilibrium.

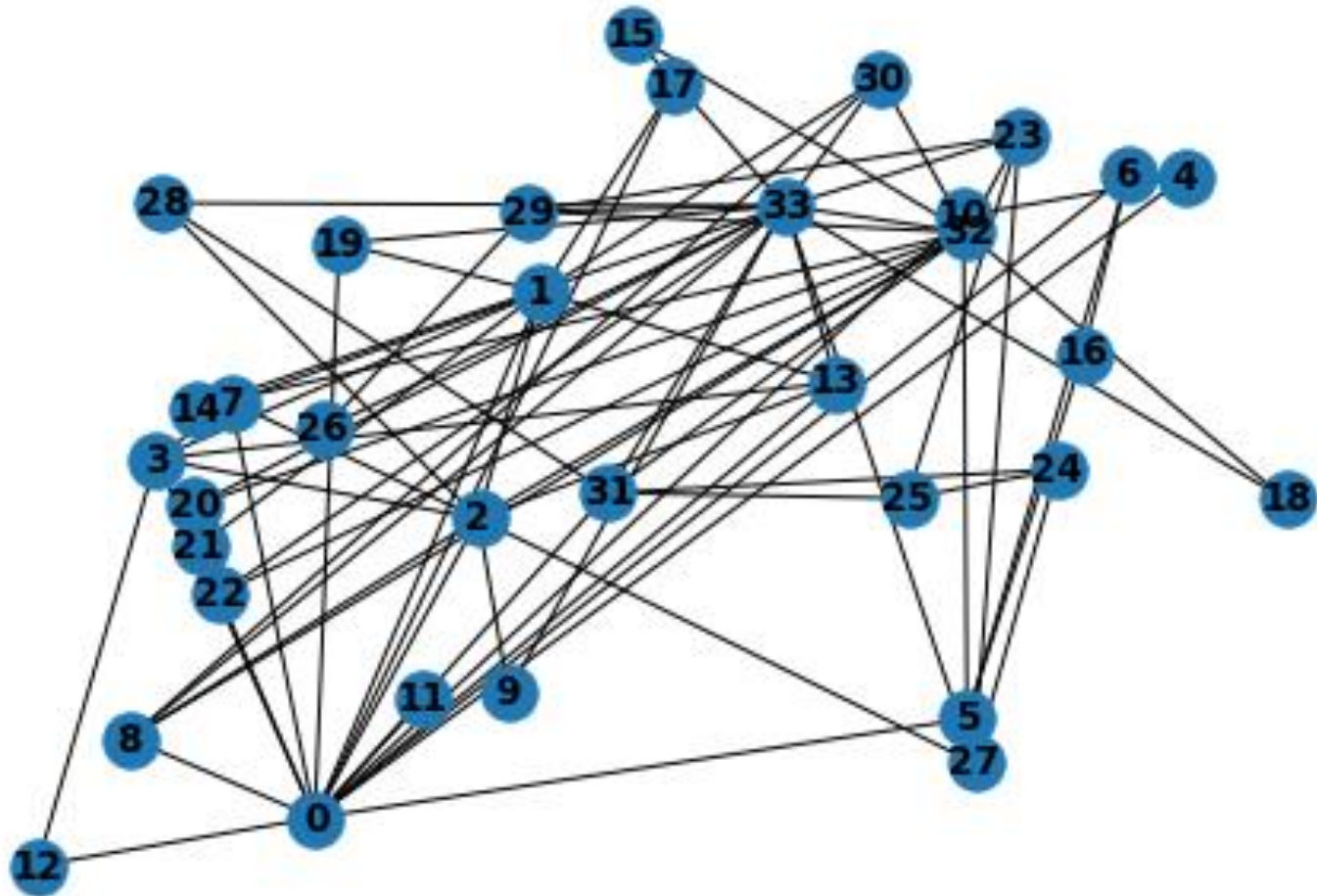
Spring Layout

Stan początkowy, iteracja 0:



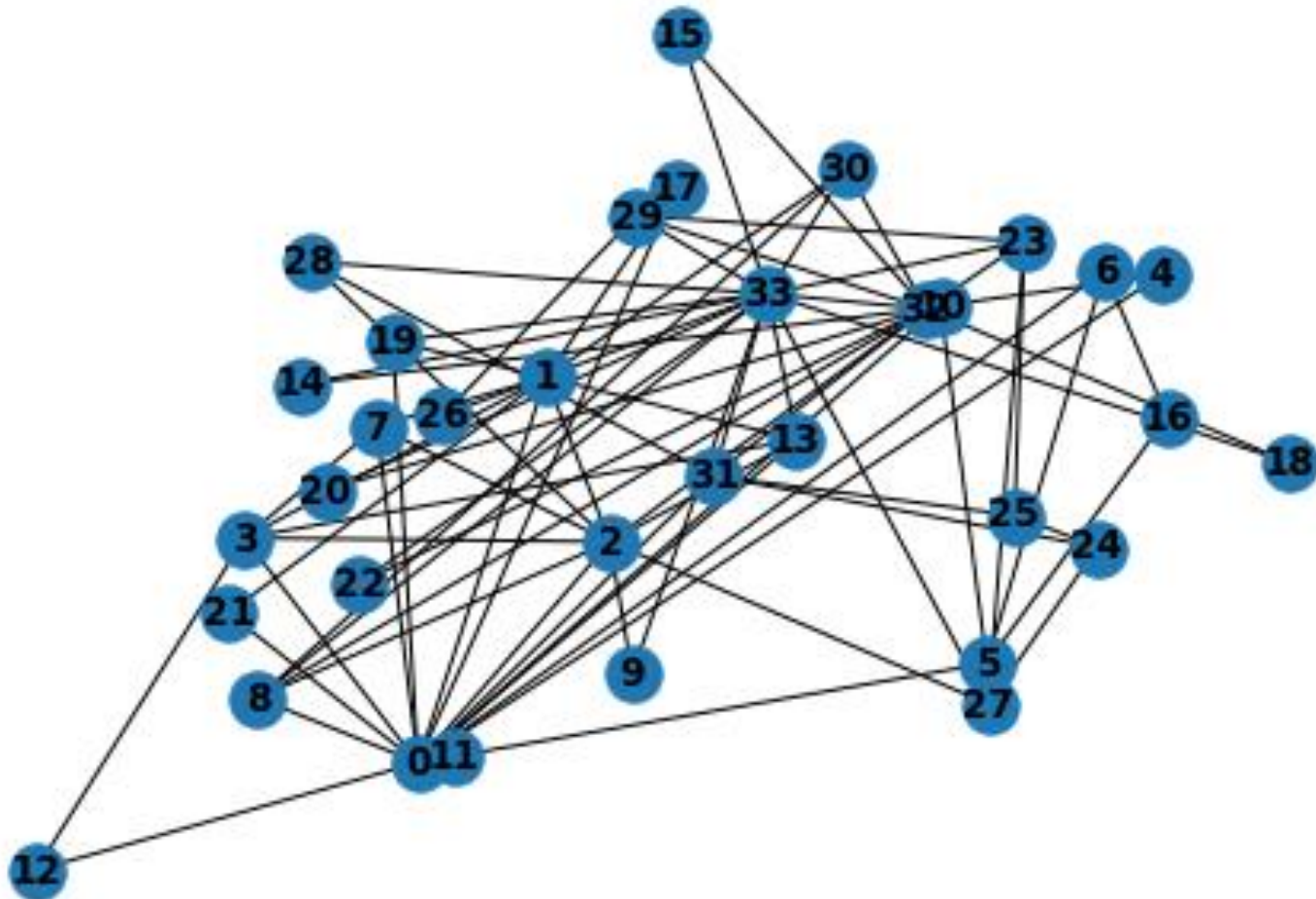
Spring Layout

Iteracja 1:



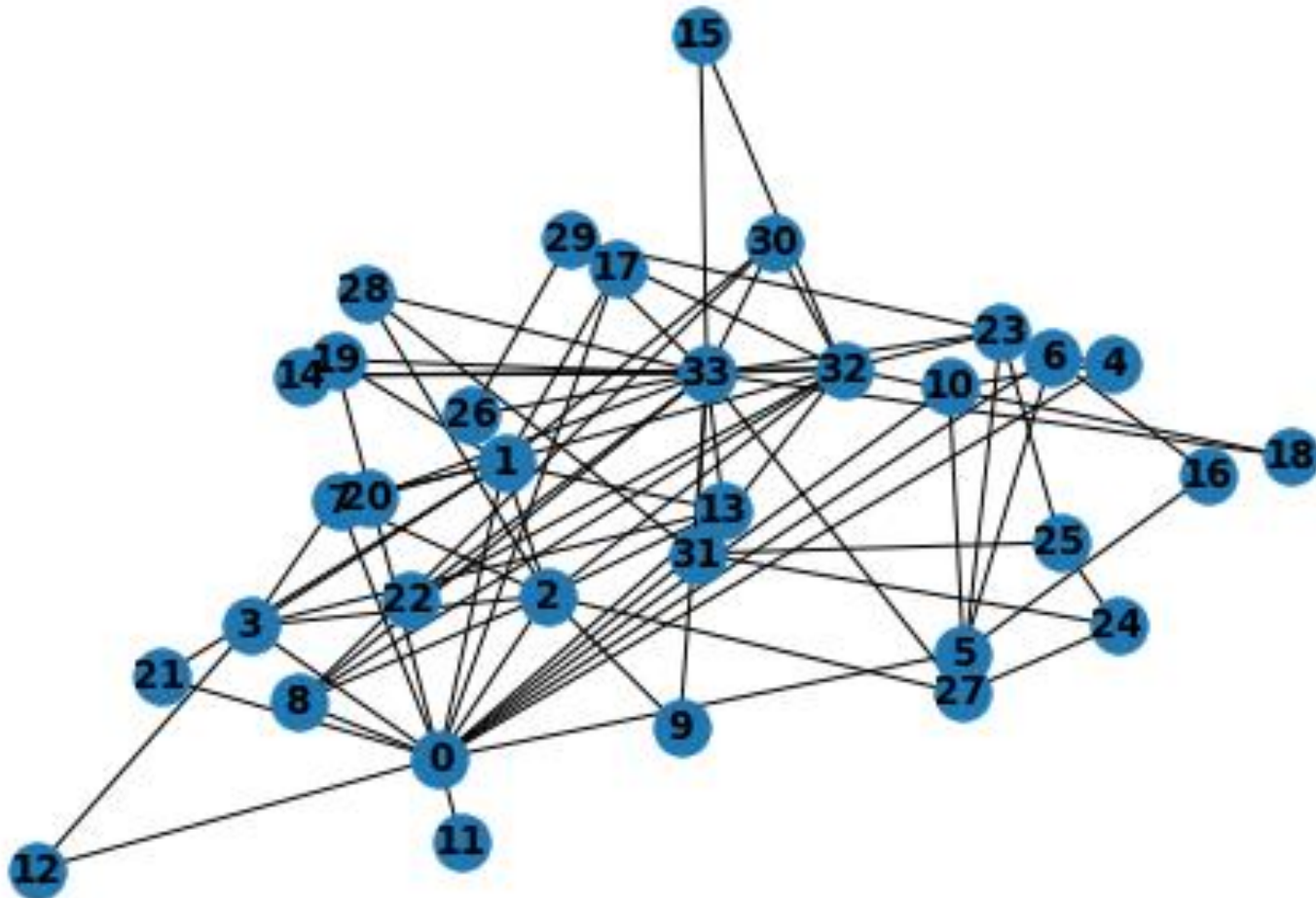
Spring Layout

Iteracja 2:



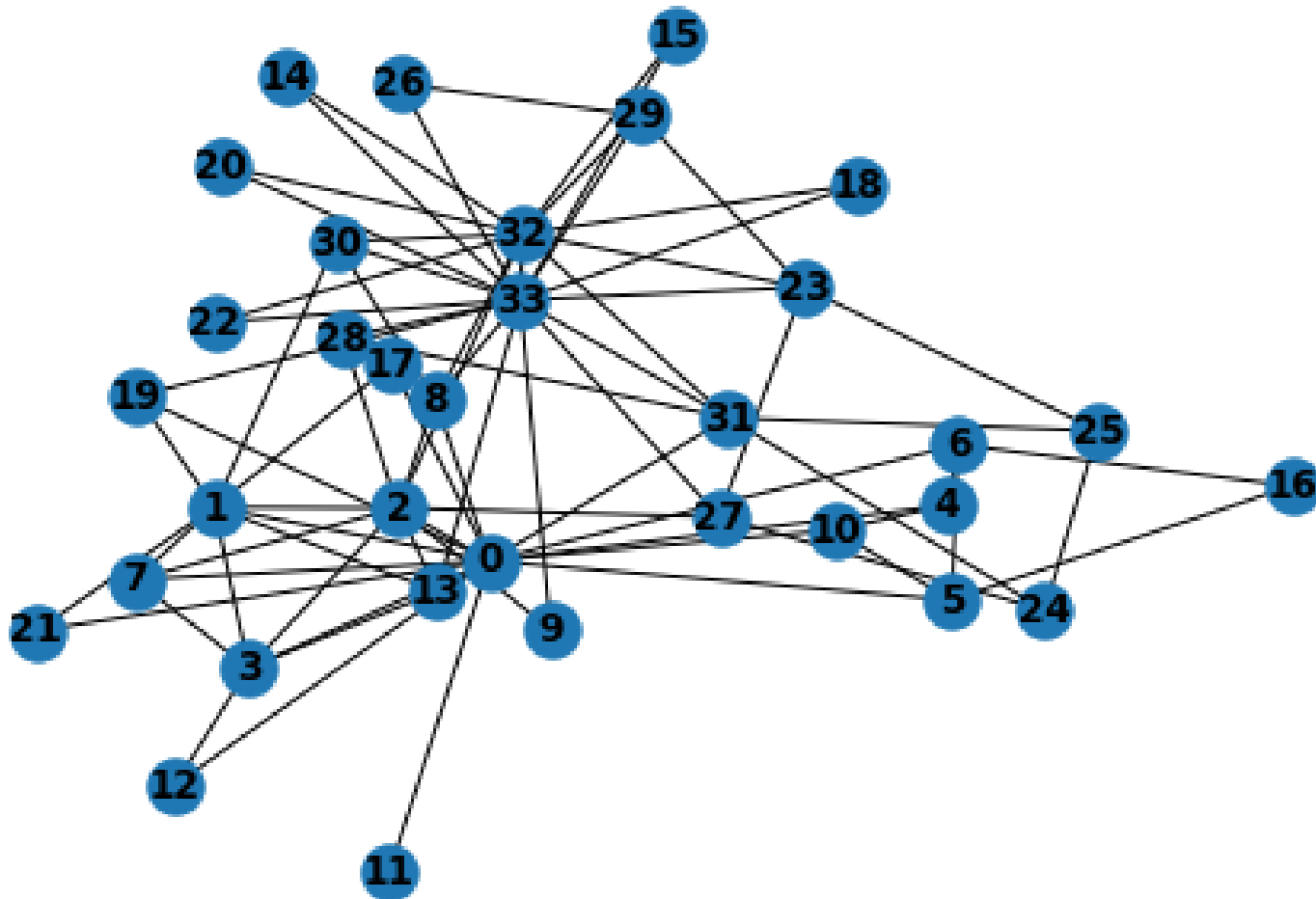
Spring Layout

Iteracja 3:



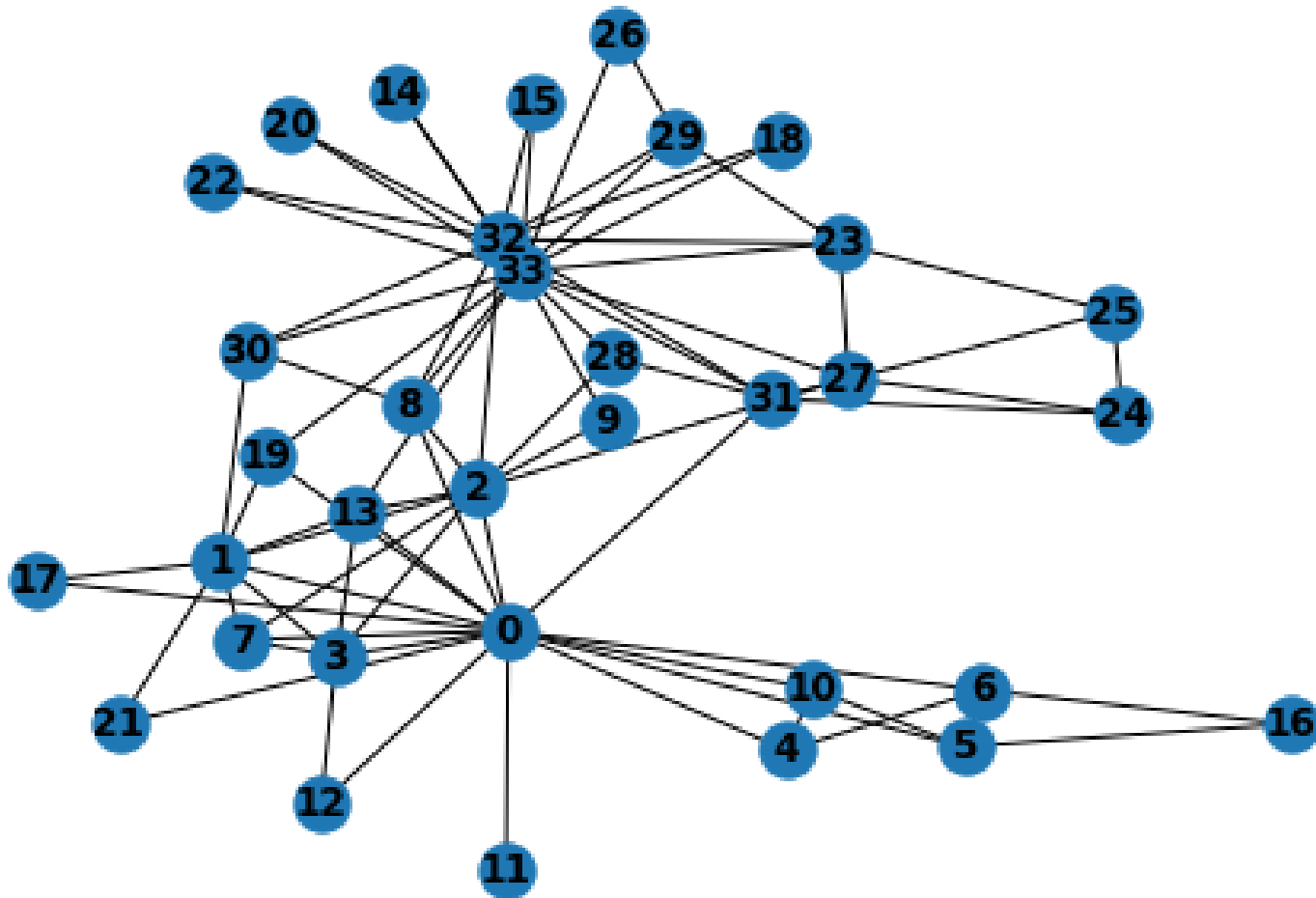
Spring Layout

Iteracja 10:



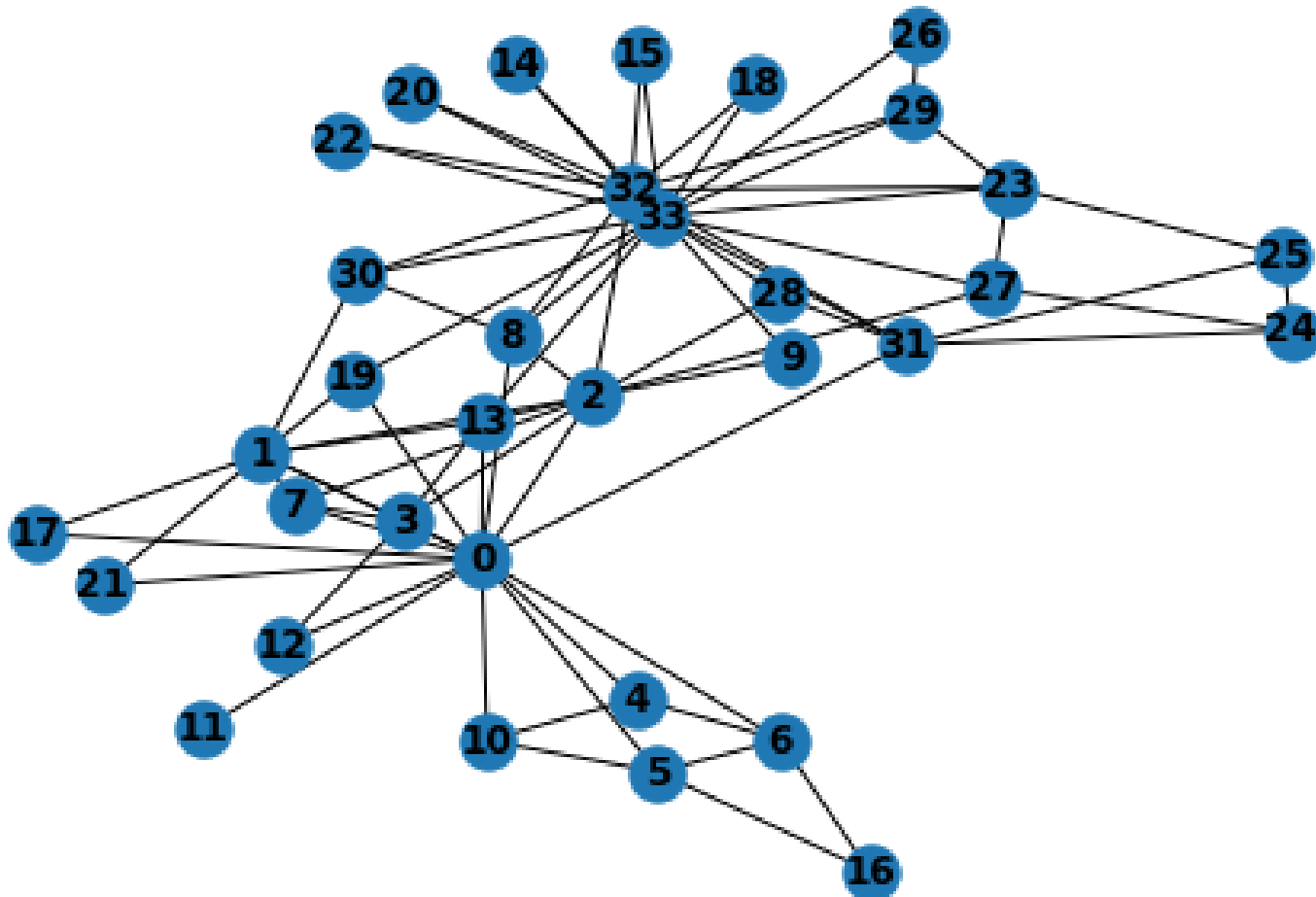
Spring Layout

Iteracja 20:



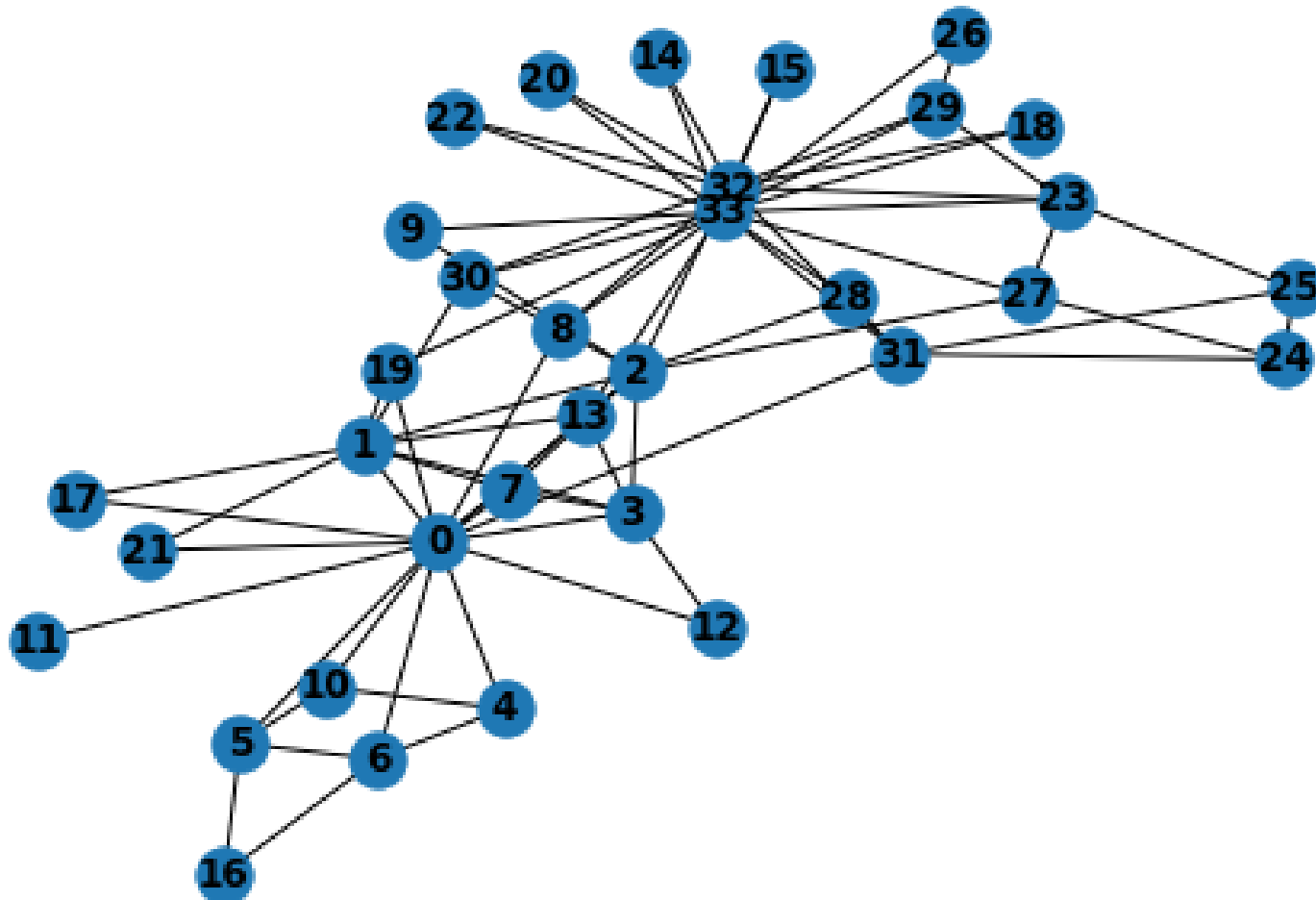
Spring Layout

Iteracja 50:



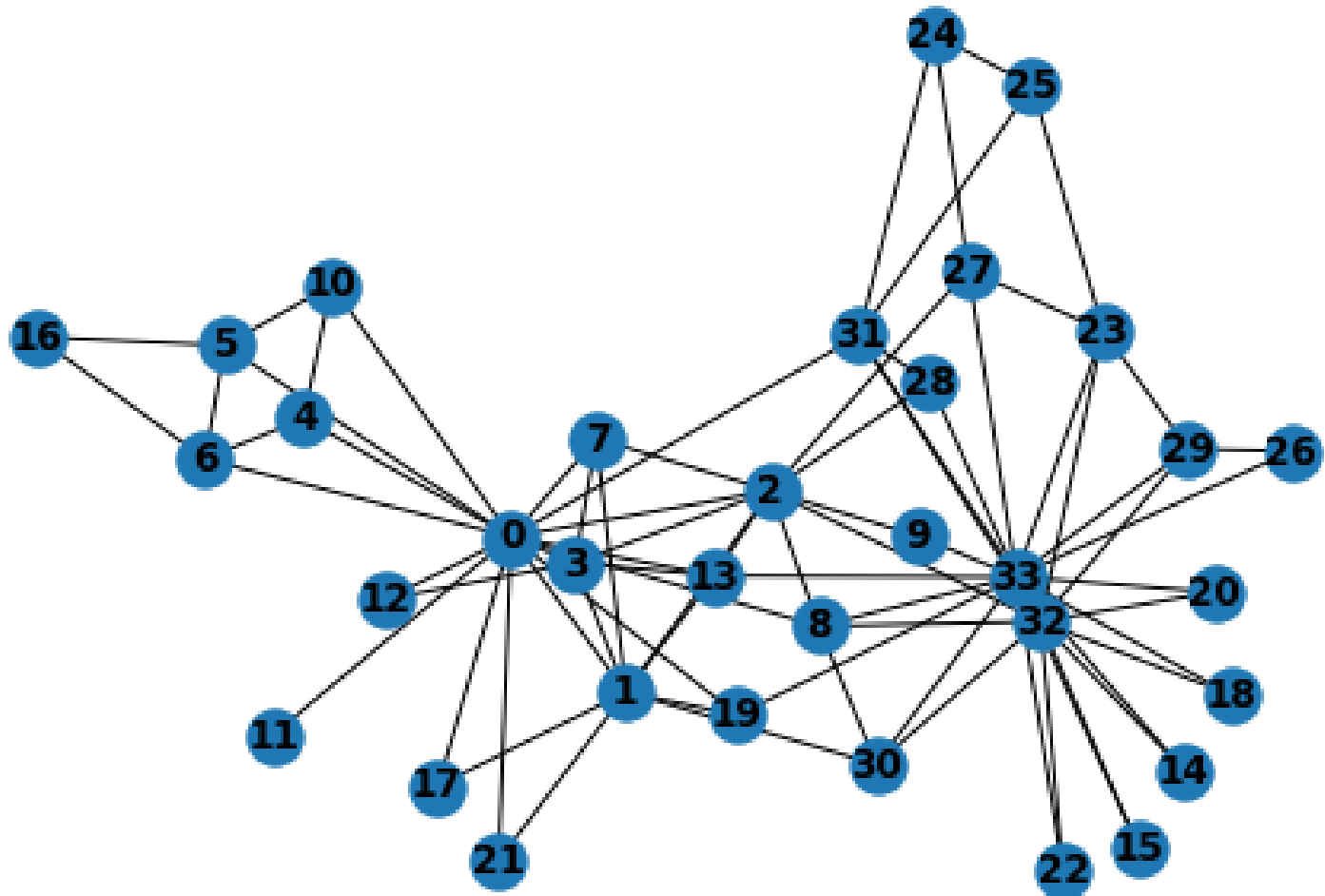
Spring Layout

Iteracja 10000:



Spring Layout

Iteracja 10000, inne ziarno:



Notebook do zabawy:

<https://github.com/Gizzio/tgsc>