



dr inż. Michał Malinowski
michal.malinowski@uth.edu.pl



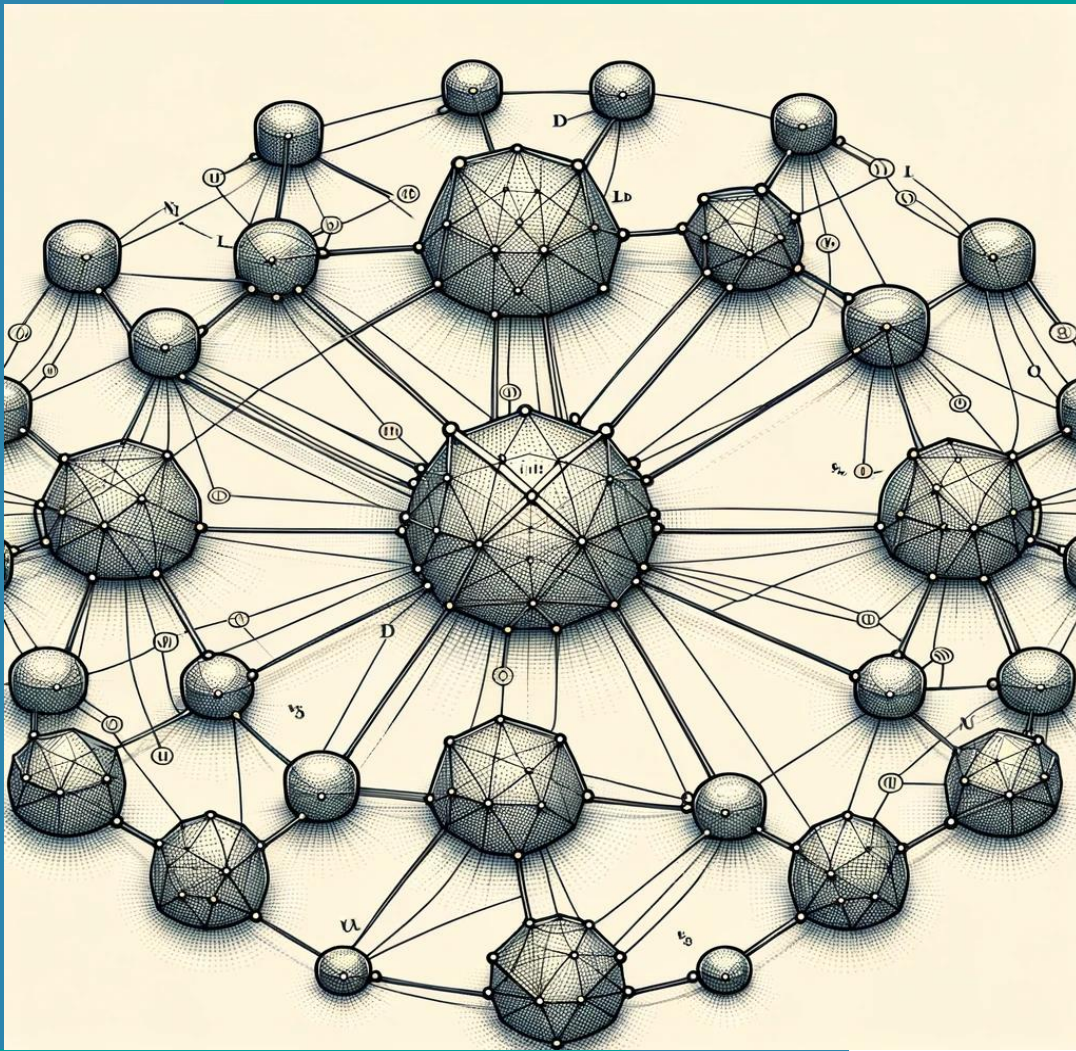
[https://www.linkedin.com/in/
michał-malinowski-malkon/](https://www.linkedin.com/in/michał-malinowski-malkon/)



Grafy

Ćwiczenie 06

ALGORYTMY
I STRUKTURY DANYCH



Zagadnienia (2h)

- [Full-Stack](#)
- [Modelowanie struktur grafowych](#)
 - [Zadanie: Modelowanie Grafu Sieci Handlowej](#)
- [Biblioteka Cytoscape.js](#)
- [Budowa struktury grafowej](#)
 - [Zadanie: Cytoscape - Drzewa binarne](#)
 - [Zadanie: Cytoscape – Sieć społecznościowa](#)

Full-Stack

Full-Stack (Deweloper)

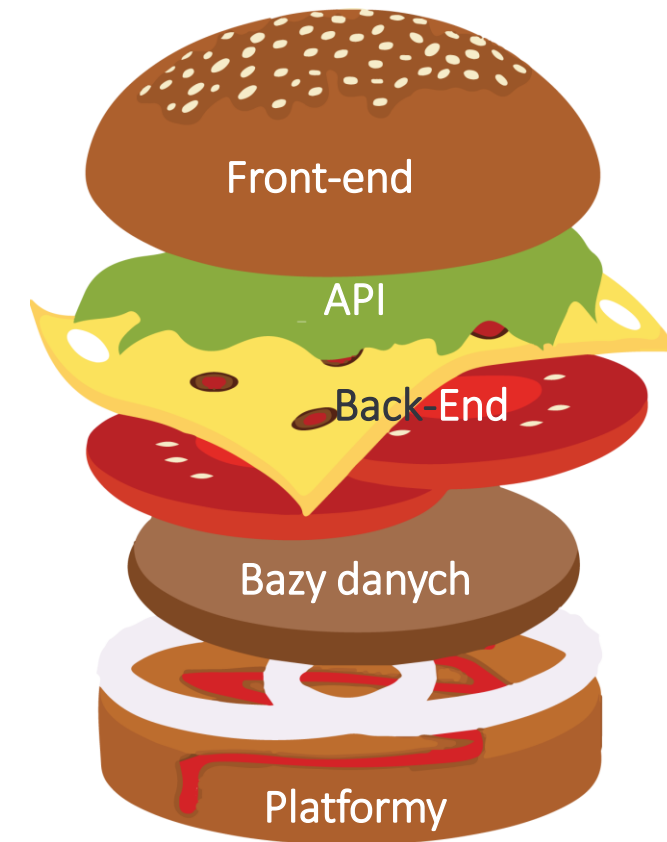
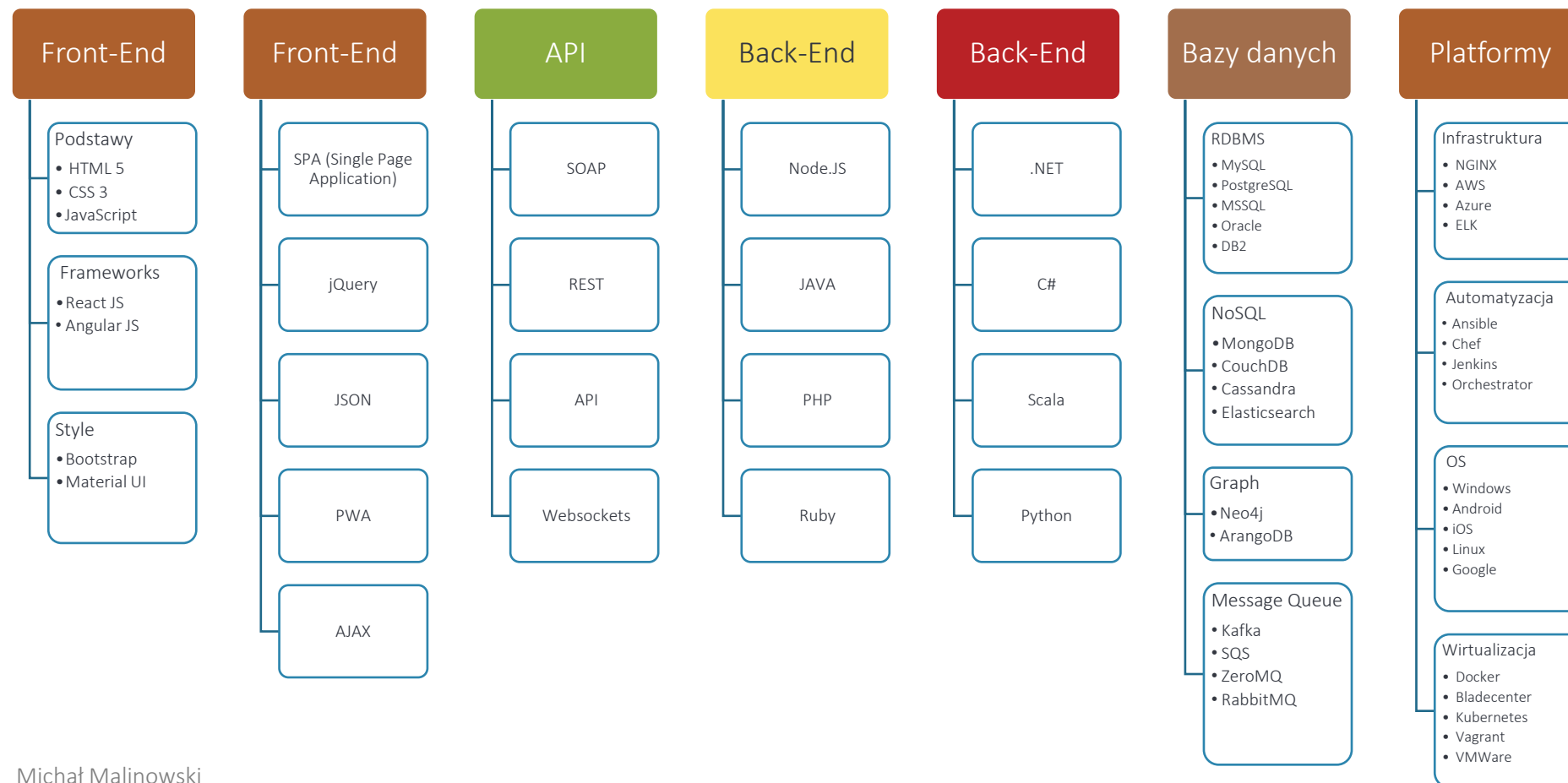
odnosi się do kompletu technologii i umiejętności niezbędnych do projektowania, implementacji oraz zarządzania całą strukturą aplikacji internetowej lub mobilnej.

Obejmuje to zarówno

- Front-End, czyli część aplikacji widoczną i dostępną dla użytkownika (interfejs użytkownika),
- Back-End, czyli część serwerową odpowiedzialną za logikę działania aplikacji, przetwarzanie danych i komunikację z bazą danych.

Programista Full-Stack jest więc w stanie samodzielnie stworzyć pełną aplikację webową lub mobilną, od zera do wersji produkcyjnej.

Full-Stack



Modelowanie struktur grafowych

Definicja

W obszarze aparatu matematycznego Grafem jest uporządkowana para:

$$G = \langle N, E \rangle$$

gdzie:

N – zbiór węzłów

E – rodzina krawędzi (rodzina w odróżnieniu od zbioru może zawierać elementy powtarzające się)

Krawędzią grafu $e \in E$ jest uporządkowana para $e = \langle n_1, n_2 \rangle$ gdzie $n_1, n_2 \in N$.

ponadto:

$N \neq \emptyset$ - graf ma co najmniej jeden węzeł

$N \cap E = \emptyset$ - zbiór węzłów i rodzina krawędzi nie mają części wspólnej.

Jeżeli rodzina E zawiera powtarzające się elementy, to takie krawędzie nazywamy równoległymi lub wielokrotnymi

Modelowanie struktur grafowych

Klasyfikacja grafów

- Graf niezorientowany (nieskierowany): jeżeli $\langle n_1, n_2 \rangle \in E$ i $\langle n_2, n_1 \rangle \in E$, to taka para jest nazywana krawędzią niezorientowaną (nieskierowaną) lub wprost krawędzią.



- Graf zorientowany (skierowany): jeżeli $\langle n_1, n_2 \rangle \in E$ i $\langle n_2, n_1 \rangle \notin E$, to taka para jest nazywana krawędzią zorientowaną (skierowaną) lub wprost krawędzią.



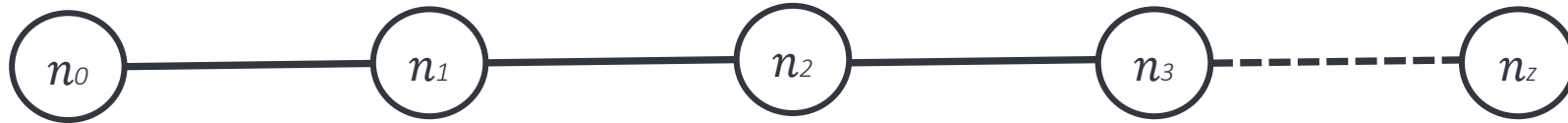
- Multigraf: graf z krawędziami równoległymi.
- Unigraf: graf bez krawędzi równoległych.



Modelowanie struktur grafowych

Ścieżka i droga w grafie

- Ścieżką w grafie: to naprzemienny ciąg węzłów i krawędzi w postaci $\{n_0 < n_0, n_1 >, n_1 < n_1, n_2 > n_2 < n_2, n_3 >, \dots, n_{(z-1)} < n_{(z-1)}, n_z >, n_z\}$, gdzie *wszystkie krawędzie są różne*.

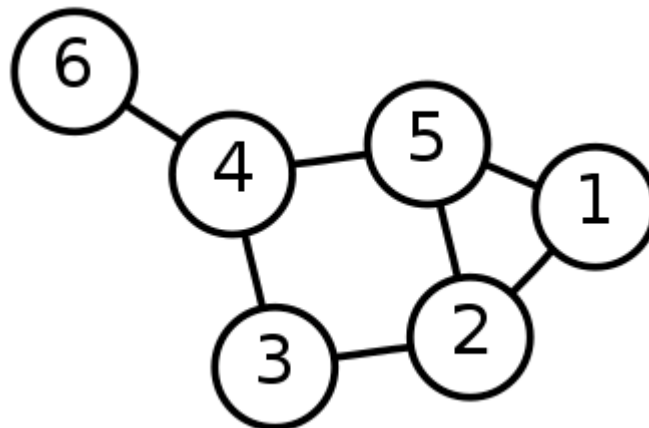


- Droga w grafie: to ścieżka, w której *wszystkie węzły są różne ewentualnie z wyjątkiem pierwszego i ostatniego węzła*.

Modelowanie struktur grafowych

Elementy Grafu

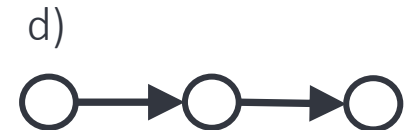
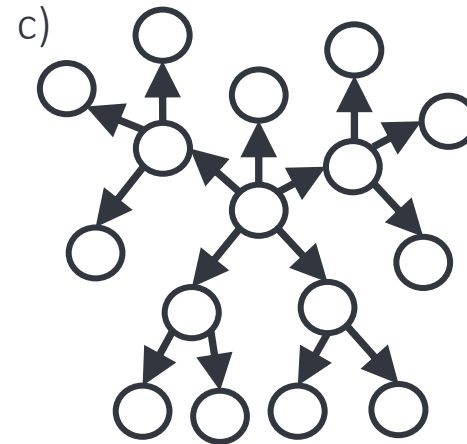
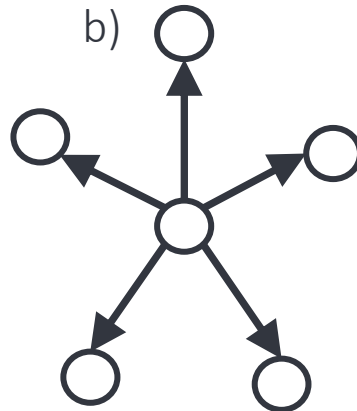
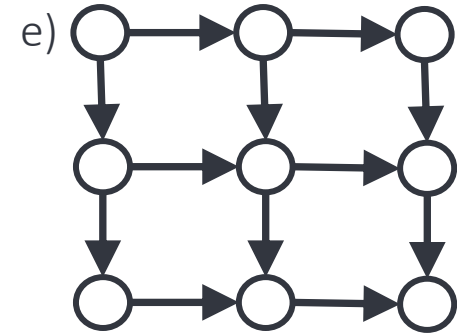
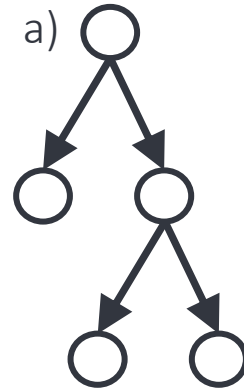
- Węzły (Wierzchołki): Podstawowe jednostki, z których składa się graf. Mogą reprezentować dowolny typ obiektu.
- Krawędzie (Łuki, Relacje): Reprezentują związki między węzłami. Mogą być skierowane (wskazujące kierunek relacji od jednego węzła do drugiego) lub nieskierowane.
- Wagi (Wartości, Atrybuty, Etykiety): Wierzchołki i krawędzie mogą zawierać atrybuty, które przechowują dodatkowe informacje.



Modelowanie struktur grafowych

Przykładowe formy grafów

- a) Drzewo np.: struktura katalogów w systemie plików
- b) Gwiazda np.: komputerowa sieć lokalna
- c) Płatek śniegu np.: sieć Internet
- d) Łańcuch np.: rzeka
- e) Siatka np.: sieć uliczna



Zadanie: Modelowanie Grafu Sieci Handlowej

Zadanie

Jesteś analitykiem systemów w lokalnej firmie spożywczej, która współpracuje z kilkoma regionalnymi dostawcami, posiada własną sieć sklepów i obsługuje licznych klientów. Twoim zadaniem jest zamodelowanie sieci dystrybucji w celu analizy przepływu produktów i identyfikacji kluczowych punktów sieci.

Zamodeluj graf (węzły, krawędzie, atrybuty) reprezentujący opisana powyżej sieć handlową składającą się z trzech dostawców, dwóch sklepów i pięciu klientów.

Zadanie: Modelowanie Grafu Sieci Handlowej

1. Zidentyfikuj Węzły:

1. Dostawcy: Każdy dostawca to węzeł.
2. Sklepy: Każdy sklep to węzeł.
3. Klienci: Każdy klient to węzeł.

2. Zdefiniuj Krawędzie:

1. Dostawy: Krawędzie od dostawców do sklepów, reprezentujące przepływ towarów.
2. Zakupy: Krawędzie od sklepów do klientów, reprezentujące transakcje.

3. Atrybuty:

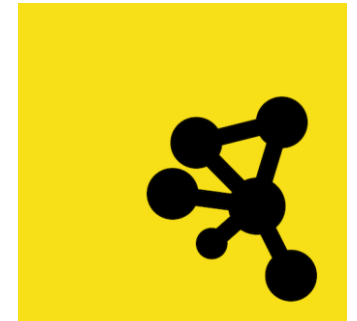
1. Na krawędziach dostaw dodaj atrybuty takie jak typ towaru, ilość i data dostawy.
2. Na krawędziach zakupów dodaj atrybuty takie jak lista zakupów, wartość transakcji i data.

Biblioteka Cytoscape.js

Cytoscape.js

to biblioteka JavaScript służąca do wizualizacji i analizy danych grafowych, czyli struktur danych w postaci węzłów i krawędzi, co może być używane do przedstawiania sieci społecznych, sieci biologicznych, map połączeń komputerowych i innych.

Pozwala użytkownikom na tworzenie interaktywnych diagramów i wizualizacji grafów w przeglądarkach internetowych.



Wizualizacja danych grafowych z węzłami i krawędziami.

Interaktywność, w tym przeciąganie węzłów, powiększanie i przewijanie.

Możliwość definiowania stylów dla węzłów i krawędzi za pomocą składni podobnej do CSS.

Obsługa różnych algorytmów układania grafu, które automatycznie pozycjonują węzły.

Możliwość rozszerzenia funkcjonalności za pomocą dodatków.

Budowa struktury grafowej

```
<!DOCTYPE html>
<html>
<head>
  <title>Prosty Graf - Cytoscape.js</title>
  <script src="https://unpkg.com/cytoscape/dist/cytoscape.min.js"></script>
  <style>
    #cy {width:600px; height:600px; display:block; margin:20px auto; border:1px solid black;}
  </style>
</head>
<body>
  <div id="cy"></div>
  <script>
    (...)
    var cy = cytoscape({
      container: document.getElementById('cy'),
      elements: graphElements,
      style: graphStyle,
      layout: graphLayout
    });
  </script>
</body>
</html>
```

biblioteka

obszar prezentacji

Inicjalizacja Cytoscape

```
var graphElements = [
  { data: { id: 'a' } },
  { data: { id: 'b' } },
  { data: { id: 'ab', source: 'a', target: 'b' } }
];
var graphStyle = [
  {
    selector: 'node',
    style: {
      'background-color': 'blue',
      'label': 'data(id)'
    }
  },
  {
    selector: 'edge',
    style: {
      'width': 3,
      'line-color': 'grey'
    }
  }
];
var graphLayout = {
  name: 'grid'
};
```

elementy

węzły

krawędzie

style

węzły

krawędzie

reprezentacja

Budowa struktury grafowej

data:

Parametr	Opis
id	Unikalny identyfikator dla wężła lub krawędzi.
source	Identyfikator wężła źródłowego krawędzi (tylko dla krawędzi).
target	Identyfikator wężła docelowego krawędzi (tylko dla krawędzi).
label	Etykieta lub nazwa wyświetlana na wężle lub krawędzi.
weight	Waga krawędzi w algorytmach grafowych.
color	Kolor wężła lub krawędzi używany do stylizacji.

```
data: { id: 'ab', source: 'a', target: 'b', label: 'Krawędź AB', weight: 10 }
```

Budowa struktury grafowej

selector:

Nazwa	Opis
'node'	Wybiera wszystkie węzły.
'edge'	Wybiera wszystkie krawędzie.

```
selector: 'node'  
selector: 'edge'
```

Budowa struktury grafowej

Style dla węzłów (selector: 'node')

Właściwość	Opis	Przykład
background-color	Kolor tła węzła.	'background-color': 'blue'
shape	Kształt węzła.	'shape': 'rectangle'
border-color	Kolor obramowania węzła.	'border-color': 'black'
border-width	Szerokość obramowania węzła.	'border-width': 2
label	Tekst etykiety wyświetlany na węźle.	'label': 'data(name)'
font-size	Rozmiar czcionki tekstu na węźle.	'font-size': 14
width i height	Szerokość i wysokość węzła.	'width': 50, 'height': 50

Budowa struktury grafowej

Style dla krawędzi (selector: 'edge')

Właściwość	Opis	Przykład
line-color	Kolor linii krawędzi.	'line-color': 'gray'
width	Szerokość linii krawędzi.	'width': 3
line-style	Styl linii krawędzi (np. solid, dotted, dashed).	'line-style': 'dotted'
curve-style	Styl krzywizny krawędzi (np. bezier, straight, unbundled-bezier).	'curve-style': 'bezier'
target-arrow-shape	Kształt strzałki na końcu krawędzi (np. triangle, circle).	'target-arrow-shape': 'triangle'
target-arrow-color	Kolor strzałki na końcu krawędzi.	'target-arrow-color': 'black'
source-arrow-shape	Kształt strzałki na początku krawędzi (analogicznie do target).	'source-arrow-shape': 'circle'

Budowa struktury grafowej

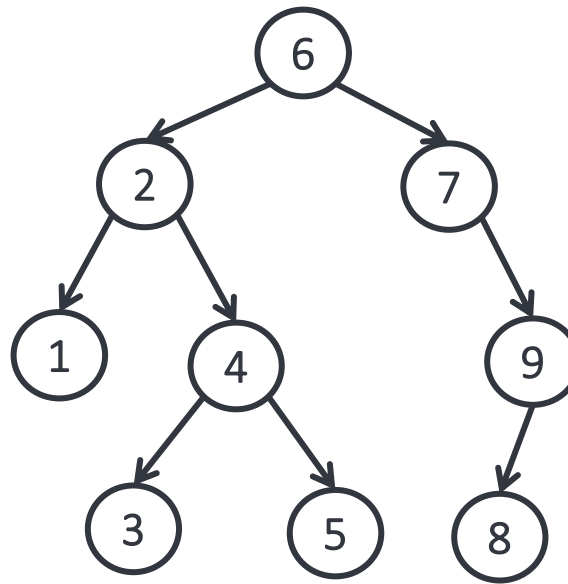
layout:

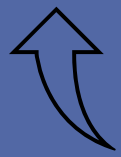
Wartość	Opis	Typ układu
grid	Rozmieszcza węzły w równomiernym siatce.	Statyczny
random	Rozmieszcza węzły w losowych pozycjach.	Statyczny
circle	Umieszcza węzły w kształcie koła.	Statyczny
concentric	Rozmieszcza węzły koncentrycznie, według określonej metryki.	Statyczny
breadthfirst	Tworzy drzewo lub hierarchię według przeszukiwania wszerz.	Hierarchiczny
cose	Symulacja fizyczna rozmieszczenia węzłów.	Siła symulacji
dagre	Dla grafów skierowanych bez cykli, układu hierarchicznie.	Hierarchiczny

Zadanie: Cytoscape - Drzewa binarne

Zadanie

Zamodeluj w Cytoscape drzewo binarne





Zadanie: Cytoscape – Sieć społecznościowa

Zadanie (2 pkt)

Zamodeluj w Cytoscape sieć społecznościową znajomych jak poniżej:

- Adam zna Gosię.
- Gosia zna Marka i Tomka.
- Marek zna Ewę.
- Tomek zna Ewę.
- Ewa zna Pawła.
- Paweł zna Adama i Olę.
- Ola zna Gosię i Marka.

Przedstawienie wyniku

- Dokument HTML
- Umieszczenie wyniku w Moodle

Termin

- Podany przy zadaniu w Moodle (około 2 tygodnie)

