

GRAFY

Piotr Śmigielski
SD 1 rok, AGH

Podstawowe pojęcia

Graf –

W ujęciu matematycznym: para $G(V, E)$, gdzie V – zbiór wierzchołków, E – zbiór krawędzi. Zbiór E składa się z par $\{a, b\}$ – (a, b należy do V) wyznaczających krawędź incydentną z wierzchołkami a i b .

Graf skierowany –

Graf, w którym zbiór krawędzi składa się z par uporządkowanych (a, b) , definiujących kierunek krawędzi

Ścieżka (droga) w grafie -

Ciąg wierzchołków (a, b, c, d, e, \dots) , takich, że w zbiorze krawędzi E występują krawędzie $\{a, b\}, \{b, c\}, \{c, d\}, \{d, e\}, \dots$ (lub $(a, b), (b, c), (c, d), (d, e), \dots$ dla grafu skierowanego)

Cykl w grafie -

Zamknięta ścieżka w grafie, mająca postać: $(a, b, c, d, e, \dots, a)$.

Najbardziej znane rodzaje cykli:

- Cykl Hamiltona – cykl, który przechodzi przez każdy wierzchołek grafu dokładnie raz
http://pl.wikipedia.org/wiki/Cykl_Hamiltona
- Cykl Eulera – cykl, który przechodzi przez każdą krawędź grafu dokładnie raz
http://pl.wikipedia.org/wiki/Cykl_Eulera
Patrz: problem „mostów królewieckich” - wniosek: *graf posiadający cykl Eulera musi mieć wszystkie wierzchołki stopnia parzystego*
http://pl.wikipedia.org/wiki/Zagadnienie_mostów_królewieckich

Klika -

Podgraf pełny grafu. Inaczej: podgraf, w którym wierzchołki łączą się każdy z każdym.

Multigraf (pseudograf) -

Graf, w którym występują krawędzie wielokrotne (powtarzające się) lub pętle (krawędzie łączące wierzchołek sam ze sobą)

Drzewo –

Graf spójny, acykliczny (nie posiadający cykli) gdy pominięto zostanie kierunek krawędzi.

Wybrane pojęcia opisujące struktury drzewiaste:

korzeń, węzeł, liść, poddrzewo, potomek, rodzic

Najczęściej wykorzystywane drzewa:

- Binarne

http://pl.wikipedia.org/wiki/Drzewo_binarne

- B-drzewo – struktura przechowywania danych

<http://pl.wikipedia.org/wiki/B-drzewo>

- Drzewo czerwono-czarne – binarne drzewo poszukiwań

http://pl.wikipedia.org/wiki/Drzewo_czerwono-czarne

- Drzewo AVL – zrównoważone, binarne drzewo poszukiwań

http://pl.wikipedia.org/wiki/Drzewo_AVL

Graf rozmyty (fuzzy graph) -

Rodzaj grafu, w którym zbiór wierzchołków jest zbiorem rozmytym a zbiór krawędzi jest rozmytą relacją na rozmytym zbiorze krawędzi.

http://shodhganga.inflibnet.ac.in/bitstream/10603/3593/4/04_chapter%201.pdf

Przykładowe zastosowania: Reprezentacja i studiowanie relacji w społeczności (socjologia, psychologia), reprezentacja map (geografia)

http://shodhganga.inflibnet.ac.in/bitstream/10603/6048/15/15_chapter%209.pdf

Znane trudne problemy algorytmiczne związane z grafami

Optymalne pokolorowanie grafu – (NP-trudny)

przypisanie wierzchołkom grafu liczb naturalnych tak aby żadne sąsiadujące ze sobą wierzchołki nie posiadały tej samej liczby, z wykorzystaniem najmniejszej możliwej ilości liczb (kolorów). Wielkość określająca minimalną ilość kolorów potrzebnych do optymalnego pokolorowania grafu to *liczba chromatyczna*.

http://pl.wikipedia.org/wiki/Kolorowanie_grafu

Patrz: „Twierdzenie o czterech barwach”

http://pl.wikipedia.org/wiki/Twierdzenie_o_czterech_barwach

Problem komiwojażera - (NP-trudny)

problem znalezienia najkrótszego Cyklu Hamiltona w pełnym grafie ważonym

http://en.wikipedia.org/wiki/Travelling_salesman_problem

Problem chińskiego listonosza – (dla grafów mieszanych NP-trudny)

problem znalezienia minimalnego cyklu w grafie ważonym (skierowanym lub nie) przechodzącego co najmniej raz przez wszystkie krawędzie grafu.

http://pl.wikipedia.org/wiki/Problem_chińskiego_listonosza

Znajdowanie klik w grafie nieskierowanym

Problem dzieli się na kilka innych:

- znajdowanie maksymalnej (w sensie zbioru wierzchołków) klik
- znajdowanie maksymalnej (w sensie sumy wag krawędzi) klik w grafie ważonym
- znajdowanie wszystkich maksymalnych klik w grafie
- sprawdzenie czy w grafie występuje klika większa niż k wierzchołków (problem decyzyjny)

Problem decyzyjny jest *NP-zupełny*

http://en.wikipedia.org/wiki/Clique_problem

Wybrane Algorytmy Grafowe

Alg. Dijkstry znajdowania najkrótszej ścieżki -

algorytm zachłanny. Wyszukuje najkrótszą ścieżkę między wybraną parą wierzchołków.

Złożoność obliczeniowa (z wykorzystaniem kopca jako kolejki przetw. krawędzi): $O(E \log V^3)$

http://pl.wikipedia.org/wiki/Algorytm_Dijkstry

Alg. Floyda-Warshalla znajdowania najkrótszych ścieżek -

wyszukuje najkrótsze ścieżki między wszystkimi parami wierzchołków w grafie

Złożoność obliczeniowa: $O(|V|^3)$

http://pl.wikipedia.org/wiki/Algorytm_Floyda-Warshalla

Alg. Kruskala znajdowania minimalnego drzewa rozpinającego

przykład algorytmu „zachłannego”. Wyszukuje krawędzie wchodzące w skład minimalnego drzewa rozpinającego dla grafu ważonego, nieskierowanego.

Złożoność obliczeniowa (przy założeniu, że na wejściu zbiór krawędzi został posortowany po wagach): $O(E \log V^3)$

http://pl.wikipedia.org/wiki/Algorytm_Kruskala

Grafy w narzędziach programistycznych

Boost Graph Library (BGL) – **biblioteka języka C++** oferująca zestaw algorytmów grafowych wraz z metodami rysowania grafów. Najważniejsze algorytmy ujęte w bibliotece: Dijkstra (najkrótsza ścieżka), Bellman-Ford (najkrótsza ścieżka), Kruskall (minimalne drzewo rozpinające), Znajdowanie Spójnych Składowych.

http://www.boost.org/doc/libs/1_48_0/libs/graph/doc/index.html

Graph-Tool – **biblioteka języka Python**, napisana w C++, oparta na *Boost Graph Library* (zawiera większość jej funkcjonalności)

<http://graph-tool.skewed.de/>

Python-Graph – **biblioteka języka Python**. Posiada zbliżoną listę zaimplementowanych algorytmów jak *Boost Graph Library*. Umożliwia eksport danych do narzędzia *Graphviz* z wykorzystaniem języka *DOT*.

<https://code.google.com/p/python-graph/>

Graphviz – **narzędzie wizualizacji grafów**. Wykorzystuje język *DOT* do definiowania formatu i treści wizualizowanego grafu.

<http://www.graphviz.org/Home.php>

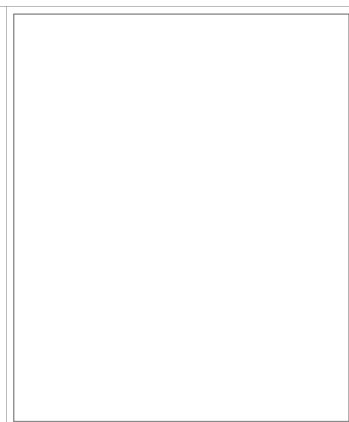
DOT – **język opisu grafu**.

http://en.wikipedia.org/wiki/DOT_graph_description_language%29

<http://www.graphviz.org/content/dot-language>

Przykład wykorzystania:

```
graph graphname {
    // This attribute applies to the graph itself
    size="1,1";
    // The label attribute can be used to change the label of a node
    a [label="Foo"];
    // Here, the node shape is changed.
    b [shape=box];
    // These edges both have different line properties
    a -- b -- c [color=blue];
    b -- d [style=dotted];
}
```



Grappa – **narzędzie wizualizacji grafów w aplikacjach Java**. Wykorzystuje język *DOT*.

<http://www2.research.att.com/~john/Grappa/>

JGraph i *JGraphT* – narzędzia do (odpowiednio) rysowania grafów, wykonywania algorytmów grafowych w języku Java.

<http://jgraph.org/visualizations.html>

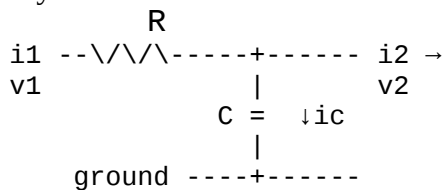
<http://jgraph.org/>

Niektóre wyspecjalizowane typy grafów

Bond Graphs – rodzaj grafów służących reprezentacji dynamicznych systemów, wewnątrz których zachodzą procesy między komponentami (na przykład układy elektryczne).

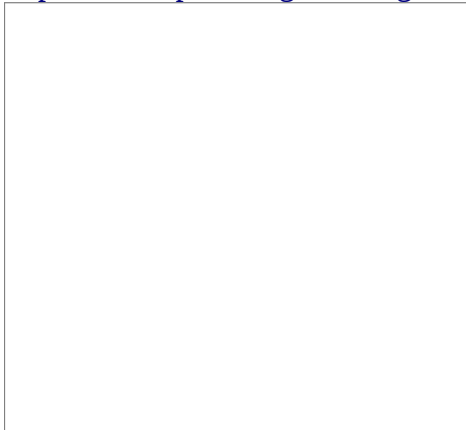
http://en.wikipedia.org/wiki/Bond_graph

Przykład:



Grafy Masona – grafy służące do reprezentacji przepływu sygnałów. Wykorzystywany m.in. do opisu stanów i zmian między stanami układów dynamicznych

http://en.wikipedia.org/wiki/Signal-flow_graph



Signed Directed Graphs (SDG)

Cognitive Graphs (Cognitive Map) – [zagadnienie biologiczne, kognitywistyczne] – sposób w jaki człowiek przechowuje wiedzę na temat otaczającej przestrzeni. Występują w niej następujące elementy składowe: drogi, węzły, punkty orientacyjne, obszary, krawędzie. Badania nad tego rodzaju strukturami w ludzkim umyśle prowadzone są w oparciu o sposób odtwarzania przetwarzanych przez umysł informacji przestrzennych.

http://en.wikipedia.org/wiki/Cognitive_map

<http://onlinelibrary.wiley.com/doi/10.1002/hipo.450010306/abstract>

Ishikawa Diagram (Fishbone Diagram) – diagram służący do graficznej reprezentacji zdarzenia i przyczyn, które do niego doprowadziły (lub mają doprowadzić). Służy głównie do zadań projektowych, odnajdywania przyczyn defektów (na przykład urządzeń) lub ogólniej, definiowania czynników, które doprowadziły do wystąpienia określonego efektu.

http://en.wikipedia.org/wiki/Ishikawa_diagram

