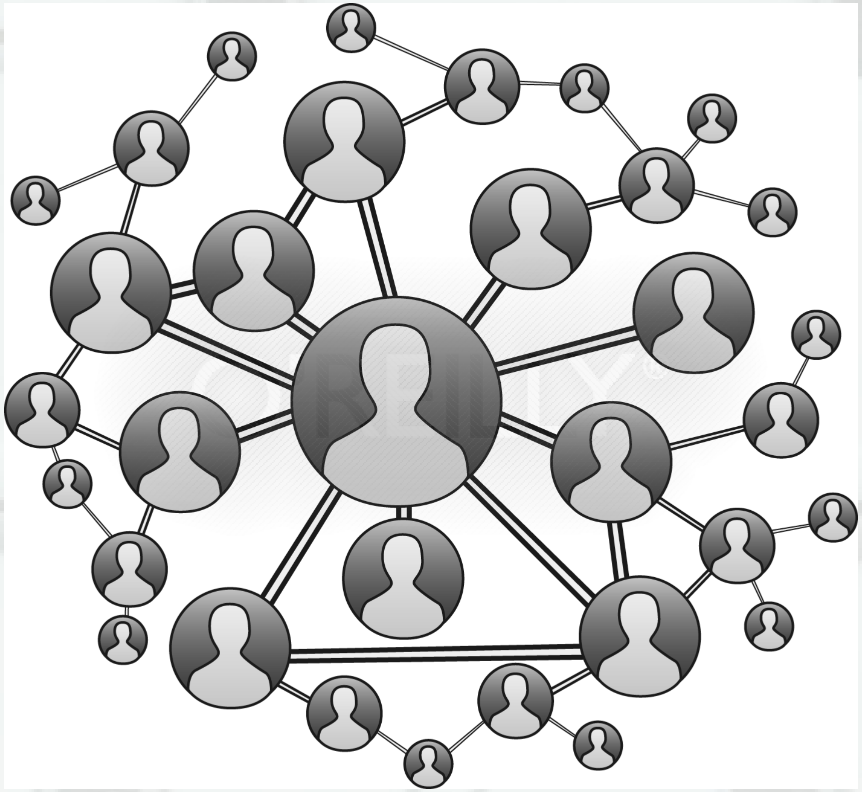
Графи



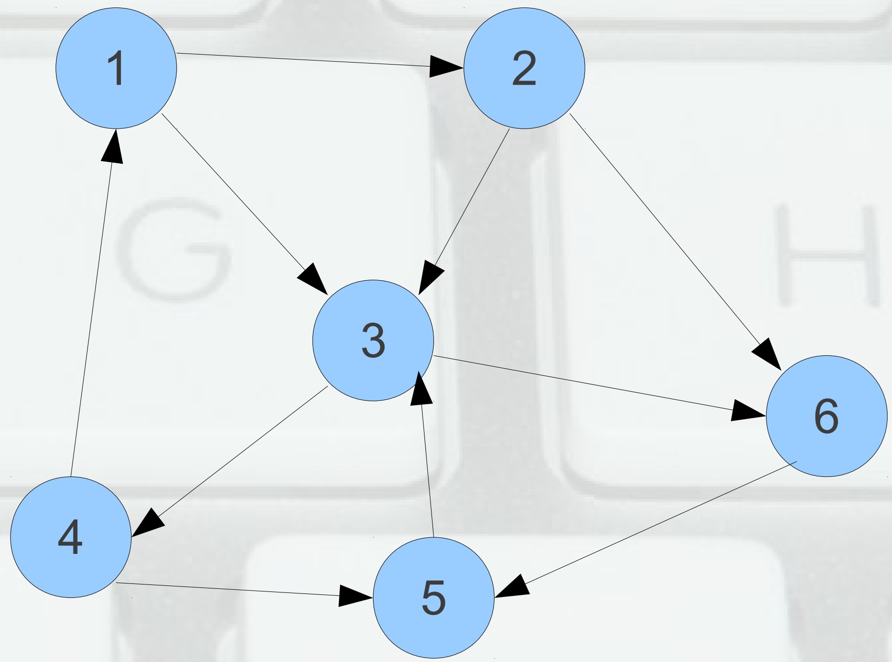
Математическа дефиниция

- Граф е наредена двойка (V, E), където V е произволно множество, а $E \subseteq V^2$
- V — върхове
- E — ребра
- $E = \emptyset$ — празен граф
- $E = V^2$ — пълен граф
- Ако редът на наредените двойки в E се пренебрегне, графът е неориентиран

Логическо описание

- Нелинейна структура, описваща обекти и връзките между тях
- Операции:
  - списък на върховете
  - списък на съседите на даден връх
  - проверка за съществуване на ребро
  - добавяне и премахване на връх
  - добавяне и премахване на ребро

Примерен граф



Етикети

- Обектите в графа могат да бъдат свързани с етикети
  - $v : V \to L$ — етикети на върховете
  - $w : E \to L$ — етикети на ребрата

Още дефиниции

- Ако (a, b) е ребро:
  - a е предшественик на b
  - b е наследник на a
- (a, a) — примка
- $d^+(a) = |\{ b \mid (a, b) \in E \}|$
- $d^-(a) = |\{ b \mid (b, a) \in E \}|$
- $d(a) = d^+(a) + d^-(a)$ — степен на a

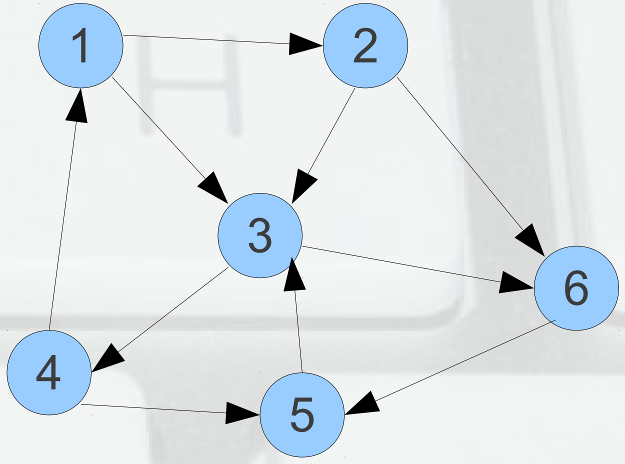
Още повече дефиниции

- Път в граф наричаме редица $v_1, v_2, ..., v_n$, за която $(v_i, v_{i+1}) \in E$
  - ако $v_1 = v_n$ — цикъл, ако $n > 1$
  - ако $v_i \neq v_j$ за $1 \leq i < j \leq n$ — ацикличен път
- Граф е цикличен, ако в него има поне един цикъл
- Граф е (слабо) свързан, ако за всеки два върха a и b има път от a до b (или от b до a)
- Път от всички ребра точно по веднъж — Ойлеров
- Път от всички върхове точно по веднъж — Хамилтонов

Матрично представяне

- Матрица на съседство

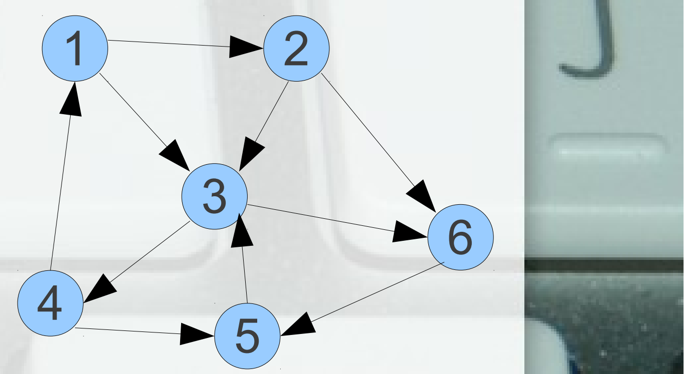
$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$



Матрично представяне

- Матрица на инцидентност

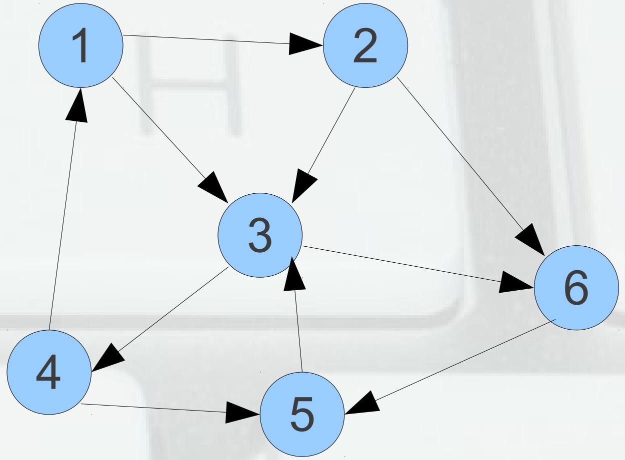
$$\begin{pmatrix} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & -1 & -1 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \end{pmatrix}$$



Свързано представяне

- Списък на наследници

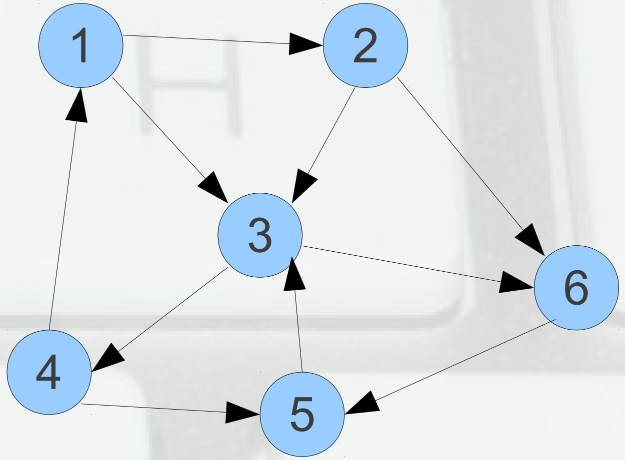
((1, (2, 3)),
  (2, (3, 6)),
  (3, (4, 6)),
  (4, (1, 5)),
  (5, (3)),
  (6, (5)))



Свързано представяне

- Списък на инцидентност

((1, 2), (2, 3), (3, 6),
(1, 3), (4, 1), (3, 4),
(5, 3), (6, 5), (4, 5),
(2, 6))



Обхождане в дълбочина

- DFS(a)
  - посети a
  - за всеки пряк наследник b на a
    - DFS(b)

Обхождане в ширина

- BFS(a)
  - маркирай a за обхождане на стъпка 1
  - за всеки връх b маркиран за обхождане на стъпка n
    - посети b
    - маркирай всички наследници на b за обхождане на стъпка n+1

Търсене на път от a до b

- Ако a == b — успех!
- Иначе:
  - добави a в пътя
  - за всеки наследник c на a, <u>който все още не е в пътя:</u>
    - търси път от c до b
    - ако има — успех!
  - ако няма път за нито един наследник на a — провал...

Всички пътища от a до b

- Ако a == b — намерен е поредният път!
- Иначе:
  - добави a в пътя
  - за всеки наследник c на a, който все още не е в пътя:
    - търси път от c до b

Проверка за цикли

- започваме от произволен връх а
- обхождаме с DFS или BFS
  - ако достигнем обратно ребро: има цикъл
  - ако обходим всички върхове: няма цикъл

Шаблон Посетител

- Visitor
  - enterVertex(T const&)
  - enterEdge(T const&, T const&)
  - exitVertex(T const&)
  - exitEdge(T const&, T const&)
- Graph
  - accept(Visitor&, T const&)

Най-кратък път от a до b

- В дълбочина — най-краткият от всички пътища
- В ширина — първият намерен път

Покриващо дърво на граф

- Обхождане в дълбочина или ширина
- Добавяне на нов връх и реброто, по което сме дошли при всяко ново посещение

Топологично сортиране

- Списък l от върхове с $d^{-s}(a) = 0$
- За всеки връх от l
  - обходи a
  - премахни a от графа
  - добави всички наследници с $d^{-}(b) = 0$ в l

Алгоритми за графи с тегла по ребрата

- Намиране на най-късите пътища от връх
  - Алгоритъм на Дийкстра
  - Алгоритъм на Флойд
- Минимално покриващо дърво
  - Алгоритъм на Прим
  - Алгоритъм на Крускал
- Максимален поток в граф
  - Алгоритъм на Форд-Фулкерсон