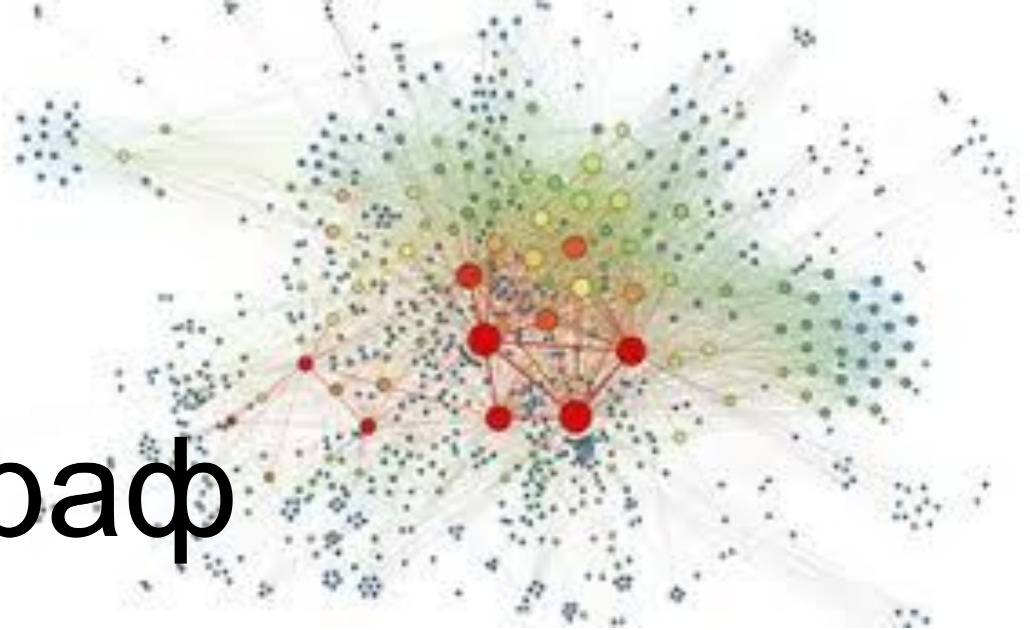


Граф

Лекция 10 по СДА, Софтуерно Инженерство
Зимен семестър 2019-2020г
д-р Милен Чечев

Какво е граф?

G(E,V) - Множество от точни и ребра

Нелинейна структура от данни съдържаща обекти-точки свързани помежду си с връзки - ребра.

Защо изучаваме алгоритми за графи?

- Хиляди практически проблеми, които се решават с тях!
- Един от най-интересните и предизвикателни раздели от компютърните науки и дискретната математика.

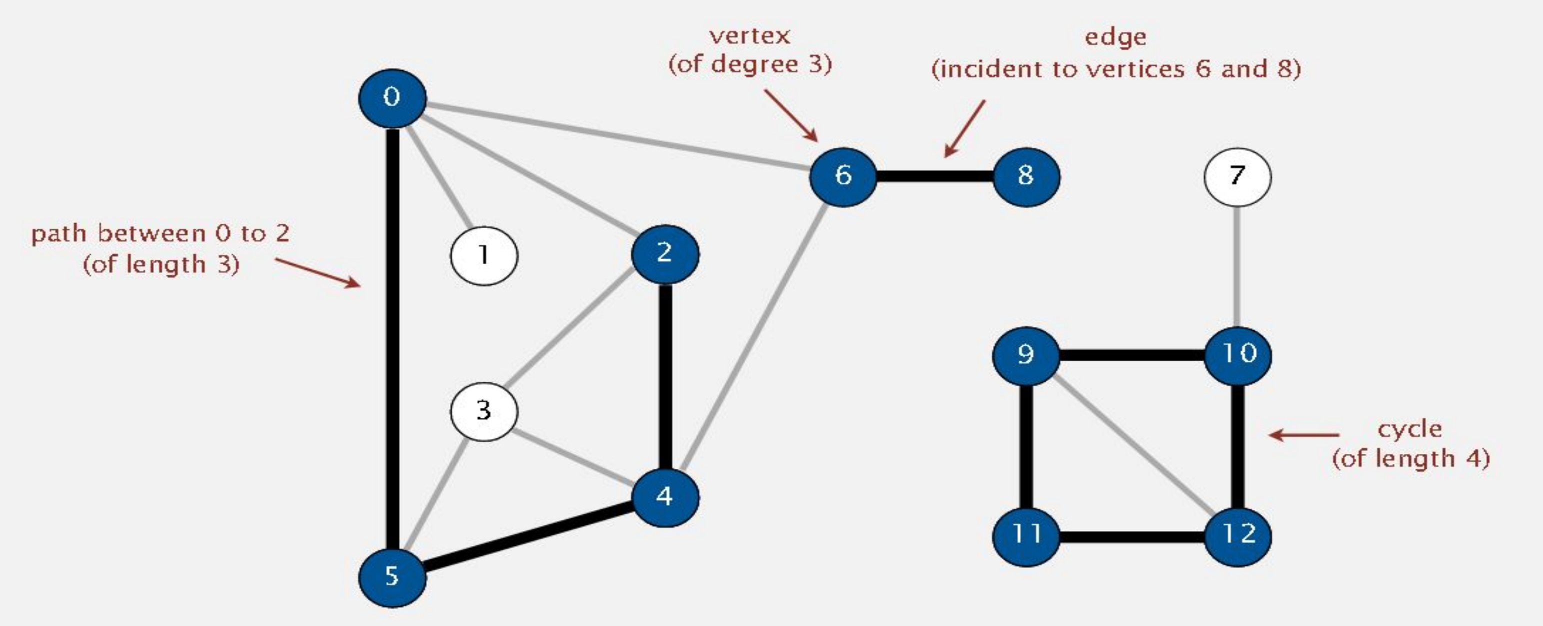
Терминология

Граф: множество от върхове свързани с ребра

Път в граф: Последователност от върхове в граф, свързани с ребро, без да се повтаря ребро.

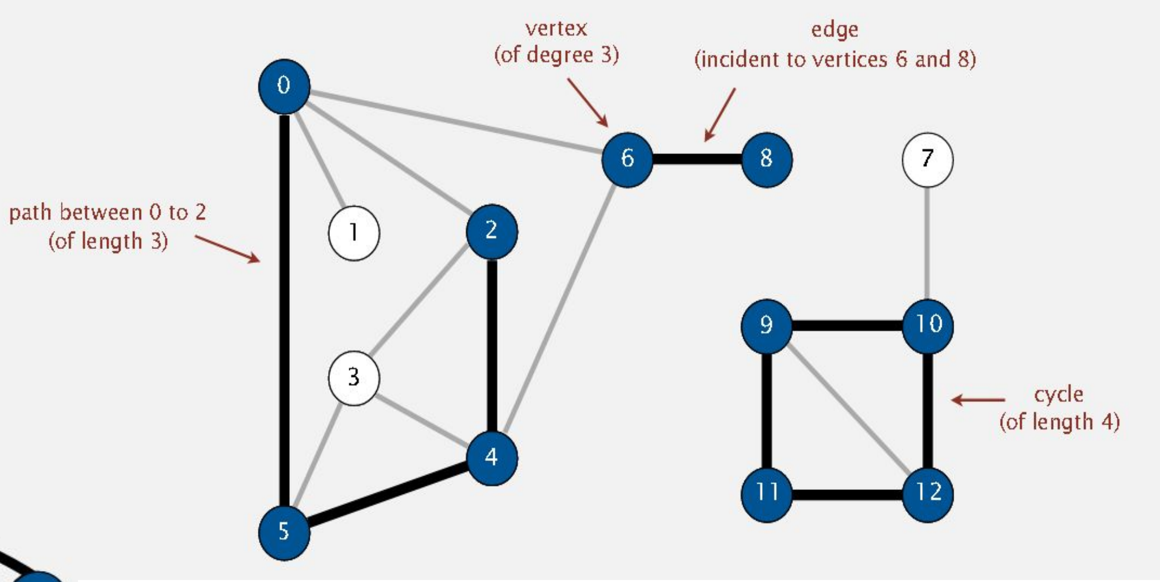
Свързаност: Два върха са свързани ако съществува път между тях

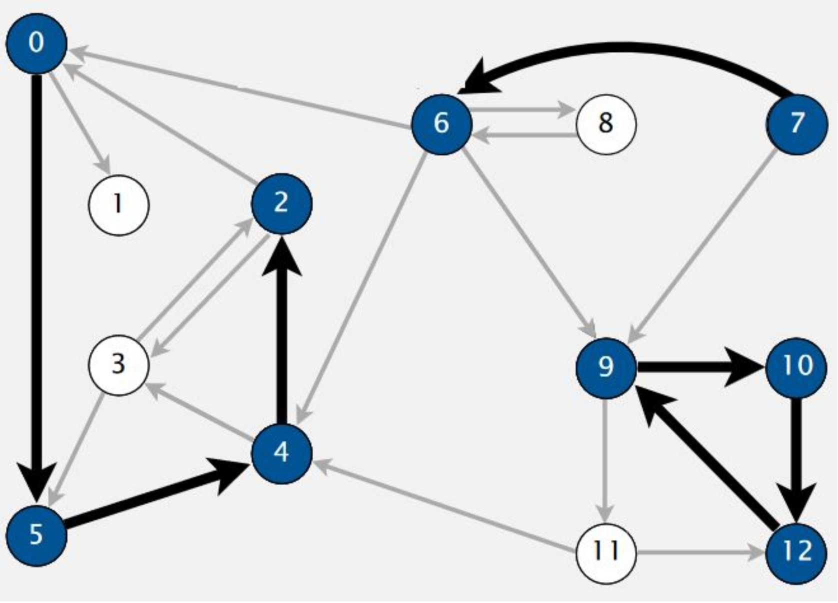
Цикъл - път с дължина повече от 1, който започва и свършва с един и същи възел



Видове граф

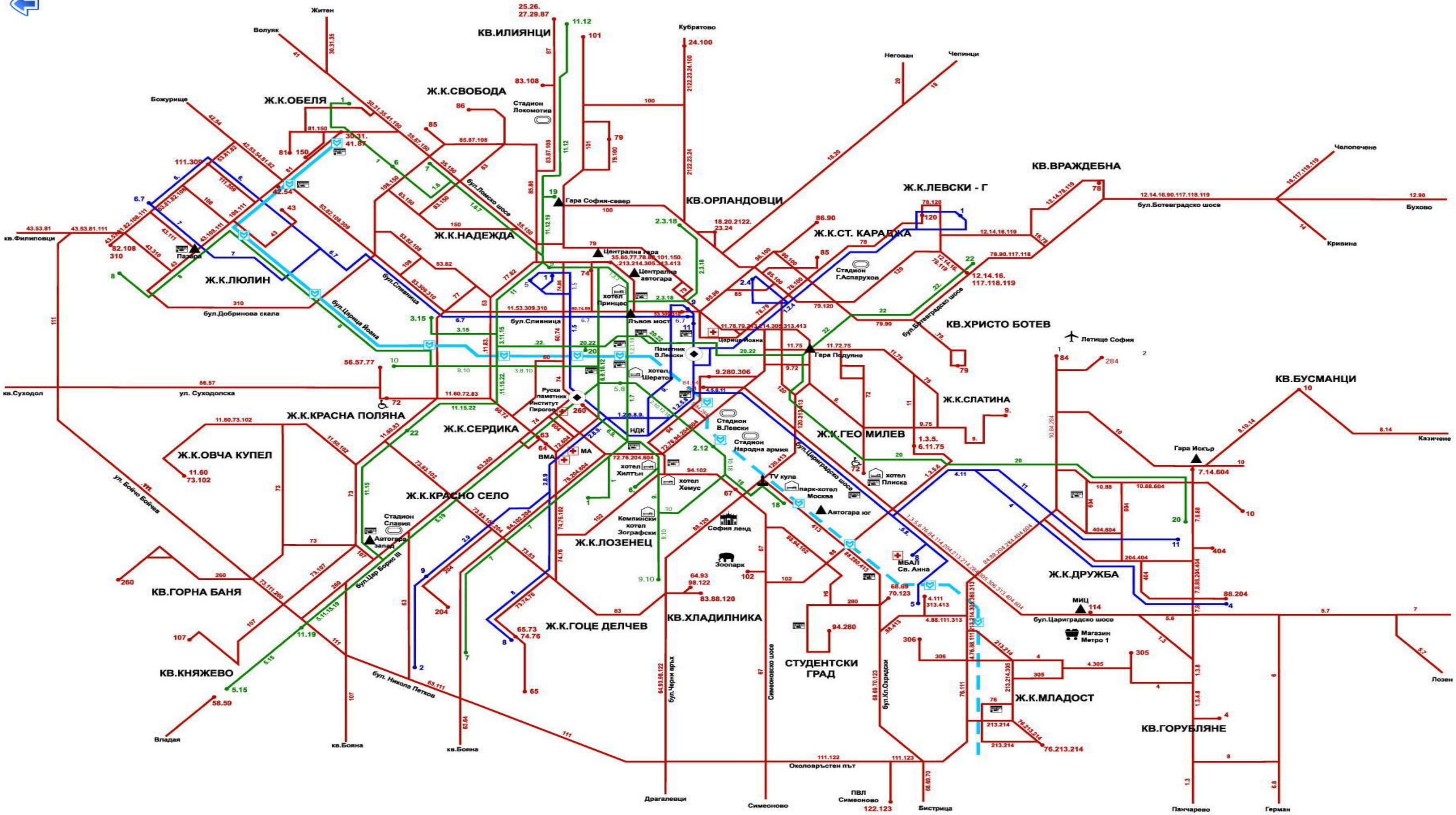
- Ненасочен(undirected)
- Насочен(directed)





Примери за графи:

https://visualgo.net/en/graphds



Житен
Волуяк
25.26.
27.29.87
КВ.ИЛИЯНЦИ
Кубратово
Негован
Чепинци
Божурище
Ж.К.ОБЕЛЯ
Ж.К.СВОБОДА
Стадион
Локомотив
Челопечене
КВ.ВРАЖДЕБНА
Ж.К.ЛЕВСКИ - Г
Гара София-север
КВ.ОРЛАНДОВЦИ
бул.Ботевградско шосе
Бухово
Кривина
кв.Филиповци
Ж.К.НАДЕЖДА
Ж.К.СТ. КАРАДЖА
Централна гара
Централна
автогара
Стадион
Г.Аспарухов
Пазара
Ж.К.ЛЮЛИН
бул.Сливница
бул.Добринова скала
хотел
Принцес
Лъвов мост
бул.Царица Йоана
КВ.ХРИСТО БОТЕВ
Летище София
Паметник
В.Левски
Гара Подуяне
хотел
Шератон
КВ.БУСМАНЦИ
кв.Суходол
ул. Суходолска
Руски
паметник
Институт
Пирогов
Ж.К.СЛАТИНА
Ж.К.КРАСНА ПОЛЯНА
Ж.К.СЕРДИКА
НДК
Стадион
В.Левски
Стадион
Народна армия
Ж.К.ГЕО МИЛЕВ
Казичене
Гара Искър
Ж.К.ОВЧА КУПЕЛ
ВМА
МА
хотел
Хилтън
хотел
Хемус
TV кула
парк-хотел
Москва
хотел
Плиска
бул.Цариградско шосе
ул. Бойчо Бойчев
Ж.К.КРАСНО СЕЛО
Кемпински
хотел
Зографски
София ленд
Автогара юг
Стадион
Славия
Автогара
запад
Ж.К.ЛОЗЕНЕЦ
Зоопарк
МБАЛ
Св. Анна
Ж.К.ДРУЖБА
бул.Цар Борис III
КВ.ГОРНА БАНЯ
МИЦ
бул.Цариградско шосе
Магазин
Метро 1
Ж.К.ГОЦЕ ДЕЛЧЕВ
КВ.ХЛАДИЛНИКА
СТУДЕНТСКИ
ГРАД
КВ.КНЯЖЕВО
бул. Никола Петков
Симеоновско шосе
бул.Черни връх
бул.Кл.Охридски
Ж.К.МЛАДОСТ
Лозен
КВ.ГОРУБЛЯНЕ
Владая
кв.Бояна
кв.Бояна
Околовръстен път
ПВЛ
Симеоново
Драгалевци
Симеоново
Бистрица
Панчарево
Герман

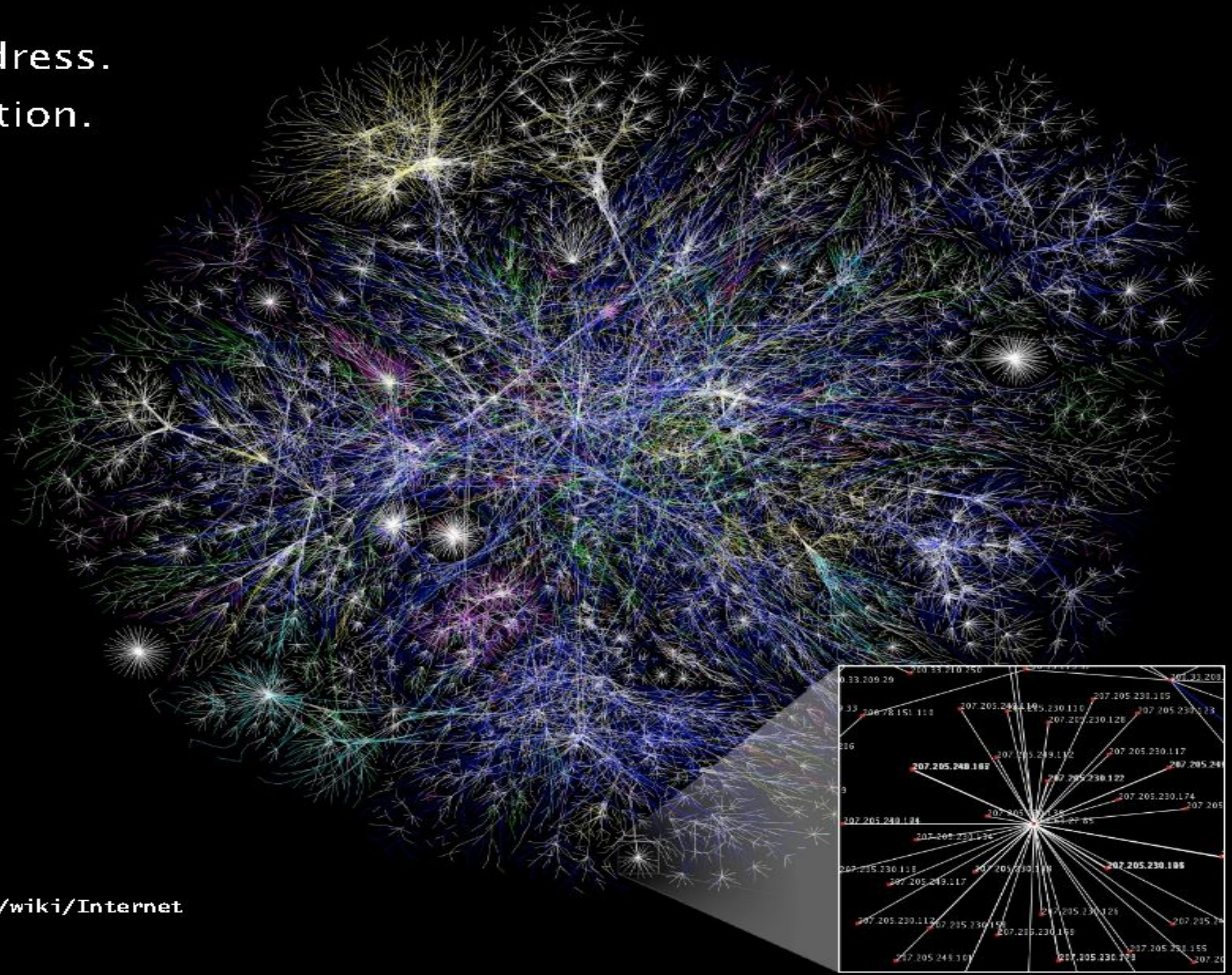

HADJI DIMITUR
KNYAGINYA MARIYA LOUISA Blvd
Central Railroad Station
SLIVNITSA Blvd
GENERAL VLADIMIR VAZOV Blvd
SOUHATA REKA
PODOUYANE
BOTEVGRADSKO SHOSSE Blvd
GEN. DANAIL NIKOLAEV Blvd
PODOUYANE Railroad Station
KONSTANTIN STOILOV Blvd
G. S. RAKOVSKI
Luvov most Sq.
HRISTO BOTEV Blvd
STEFAN STAMBOULOV Blvd
KNYAGINYA MARIYA LOUISA Blvd
OPULCHENSKA Blvd
VASSIL LEVSKI Blvd
KNYAZ AL. DONDOUKOV Blvd
MADRID Blvd
YANKO SAKUZOV Blvd
EVLOGI GEORGIEV Blvd
SITNYAKOVO
AL. STAMBOLIYSKI Blvd
SUBORNA
Sveta Nedelya Sq.
Al. Battenberg Sq.
TSAR OSVOBODITEL Blvd
Aleksandur Nevski Sq.
Vassil Levski Sq.
Narodno Subranie Sq.
SKOBELEV Blvd
DAMYAN GRUEV Blvd
MAKEDONIYA Blvd
Makedoniya Sq.
VITOSHA Blvd
Ruski pametnik Sq.
GEN. SKOBELEV Blvd
PATRIARCH EVTIMI Blvd
Orlov most Sq.
TSARIGRADSKO SHOSSE Blvd
Stadium "Vassil Levski"
BORISSOVA GRADINA
SHIPCHENSKI PROHOD Blvd
GEO MILEV
PENCHO SLAVEYKOV Blvd
PRAGA Blvd
BABA NEDELYA Blvd
National Palace of Culture
BULGARIA Blvd
CHERNI VRAH
DRAGAN TSANKOV Blvd
HRISTO SMIRNENSKI
PEYU YAVOROV Blvd
Airport


SOCIAL
NETWORK

The Internet as mapped by the Opte Project

Vertex = IP address.

Edge = connection.



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

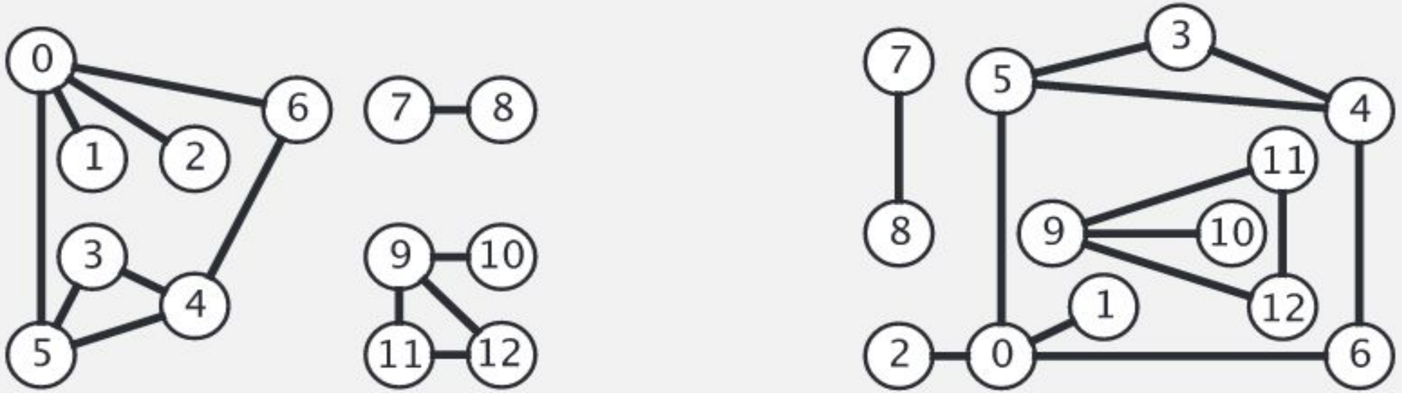
…и още…

graph	vertex	edge
communication	telephone, computer	fiber optic cable
circuit	gate, register, processor	wire
mechanical	joint	rod, beam, spring
financial	stock, currency	transactions
transportation	intersection	street
internet	class C network	connection
game	board position	legal move
social relationship	person	friendship
neural network	neuron	synapse
protein network	protein	protein–protein interaction
molecule	atom	bond

problem	description
s-t path	*Is there a path between s and t ?*
shortest s-t path	*What is the shortest path between s and t ?*
cycle	*Is there a cycle in the graph ?*
Euler cycle	*Is there a cycle that uses each edge exactly once ?*
Hamilton cycle	*Is there a cycle that uses each vertex exactly once ?*
connectivity	*Is there a path between every pair of vertices ?*
biconnectivity	*Is there a vertex whose removal disconnects the graph ?*
planarity	*Can the graph be drawn in the plane with no crossing edges ?*
graph isomorphism	*Are two graphs isomorphic?*

Graph representation

Graph drawing. Provides intuition about the structure of the graph.



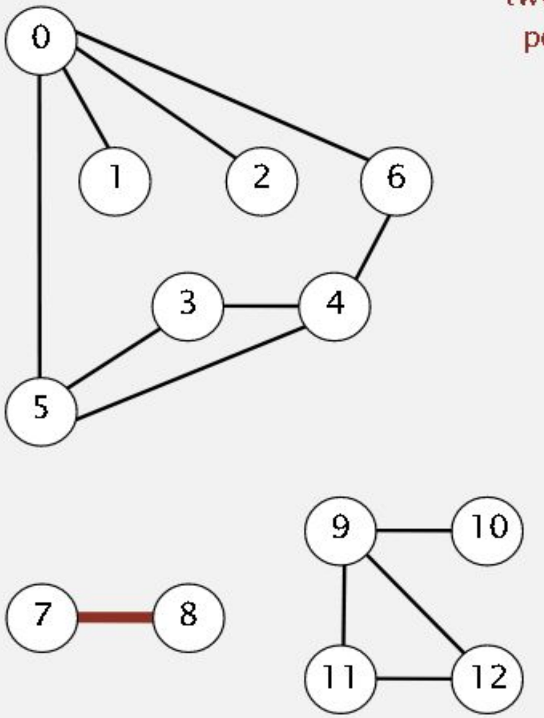
two drawings of the same graph

Caveat. Intuition can be misleading.

Graph representation: adjacency matrix

Maintain a V-by-V boolean array; for each edge v–w in graph:

`adj[v][w] = adj[w][v] = true.`



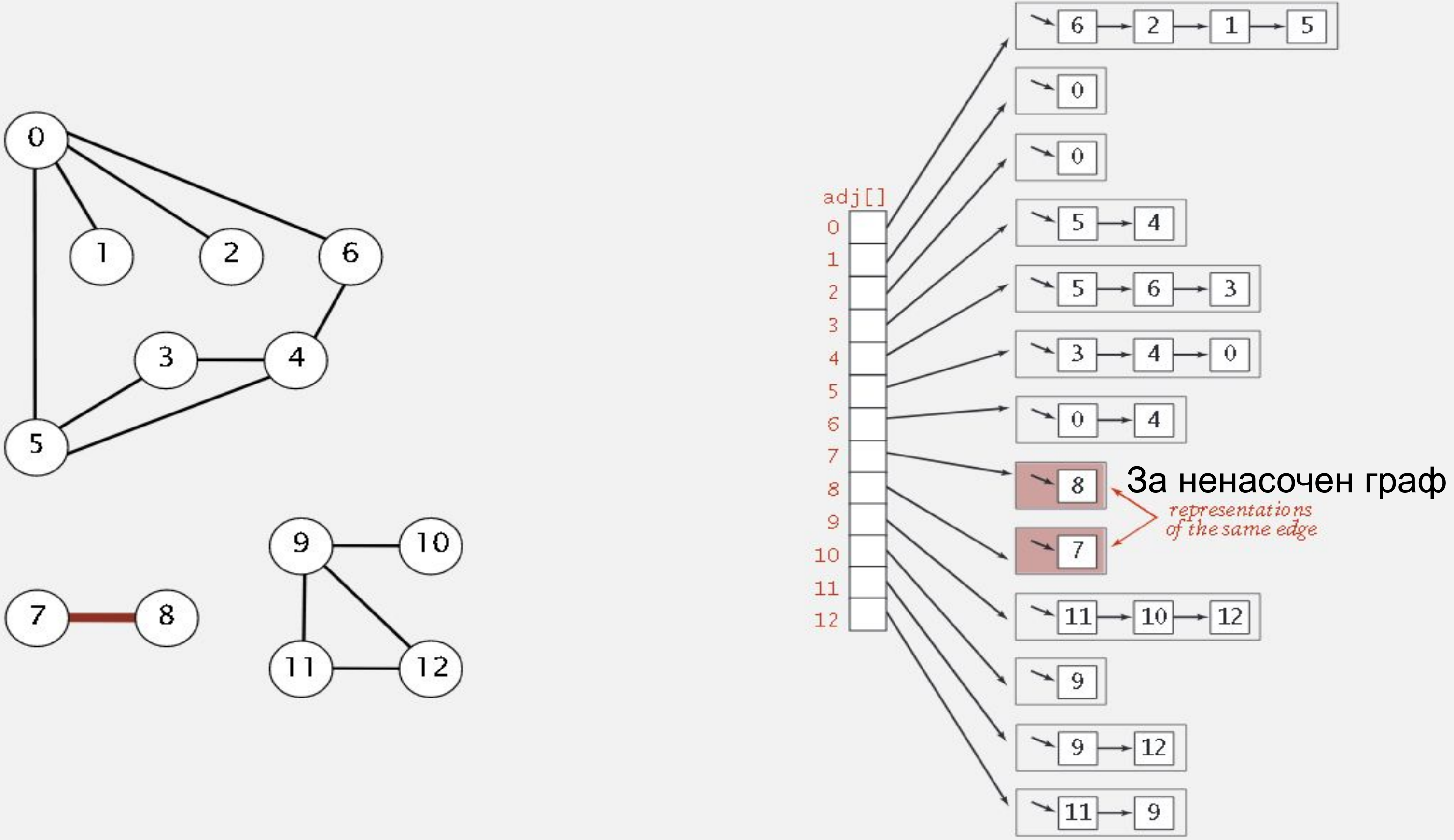
two entries per edge

(За ненасочен граф)

	0	1	2	3	4	5	6	7	8	9	10	11	12
0	0	1	1	0	0	1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	1	1	0	0	0	0	0	0	0
4	0	0	0	1	0	1	1	0	0	0	0	0	0
5	1	0	0	1	1	0	0	0	0	0	0	0	0
6	1	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	1	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	1	1	1
10	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0	0	1
12	0	0	0	0	0	0	0	0	0	1	0	1	0

Graph representation: adjacency lists

Maintain vertex-indexed array of lists.



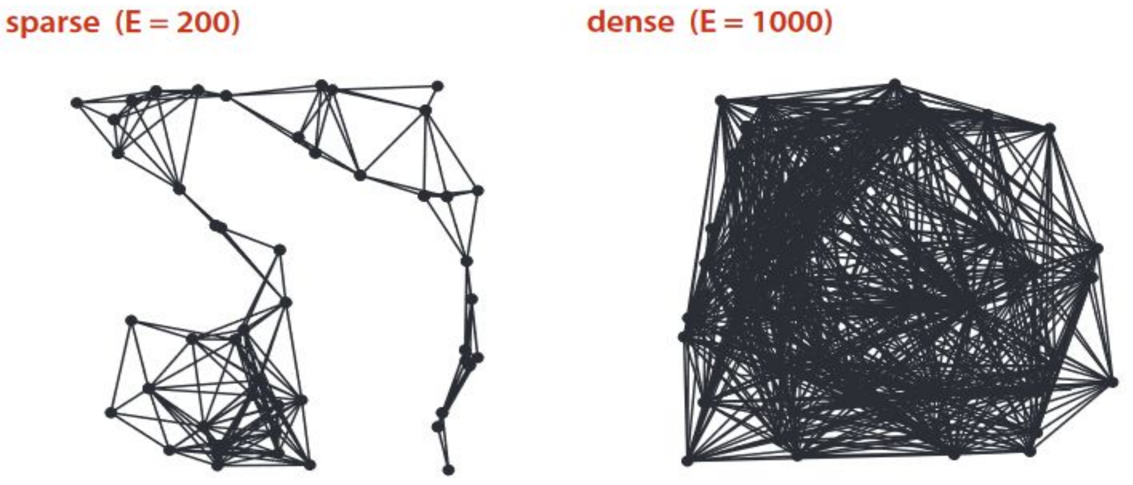
Graph representations

In practice. Use adjacency-lists representation.

- Algorithms based on iterating over vertices adjacent to v.
- Real-world graphs tend to be sparse (not dense).

proportional to V edges

proportional to V^2 edges



Two graphs (V = 50)

Сложност на различните представяния

representation	space	add edge	edge between v and w?	iterate over vertices adjacent to v?
list of edges	E	1	E	E
adjacency matrix	V^2	1 †	1	V
adjacency lists	$E + V$	1	$degree(v)$	$degree(v)$

† disallows parallel edges

Обхождане на граф

1. Обхождане в дълбочина (dfs)
2. Обхождане в ширина (bfs)

Обхождане в дълбочина

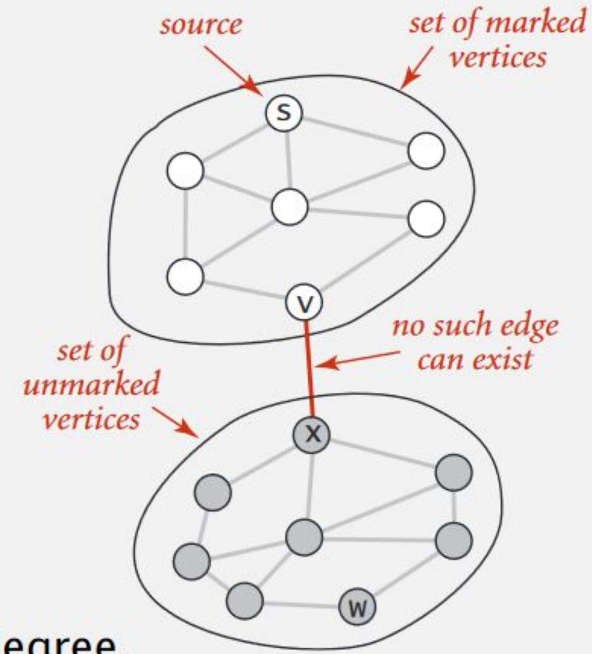
```
void dfs(int v, bool visited[], list<int> *adj; ) {
    visited[v] = true;
    // Проверка на специфично условие което се търси.

    list<int>::iterator i;
    for (i = adj[v].begin(); i != adj[v].end(); ++i){
        if (!visited[*i]){
            dfs(*i, visited,adj);
        }
    }
}
```

Depth-first search: properties

Proposition. DFS marks all vertices connected to s in time proportional to $V + E$ in the worst case.

Pf. [correctness]

- If w marked, then w connected to s (why?)
- If w connected to s, then w marked.
  (if w unmarked, then consider the last edge on a path from s to w that goes from a marked vertex to an unmarked one).



Pf. [running time]

- Each vertex is visited at most once.
- Visiting a vertex takes time proportional to its degree.

$$degree(v_0) + degree(v_1) + degree(v_2) + \ldots = 2E$$

Обхождане в ширина

```
void BFS(int s,list<int> *adj) {
	bool *visited = new bool[V];
	for(int i = 0; i < V; i++){
		visited[i] = false;
	}
	list<int> queue;
	queue.push_back(s);
	visited[s] = true;
	list<int>::iterator i;
	while(!queue.empty()) {
		s = queue.front();
		queue.pop_front();
		// Проверка на специфично условие, което се търси.
		for (i = adj[s].begin(); i != adj[s].end(); ++i) {
			if (!visited[*i]) {
				visited[*i] = true;
				queue.push_back(*i);
			}
		}
	}
}
```

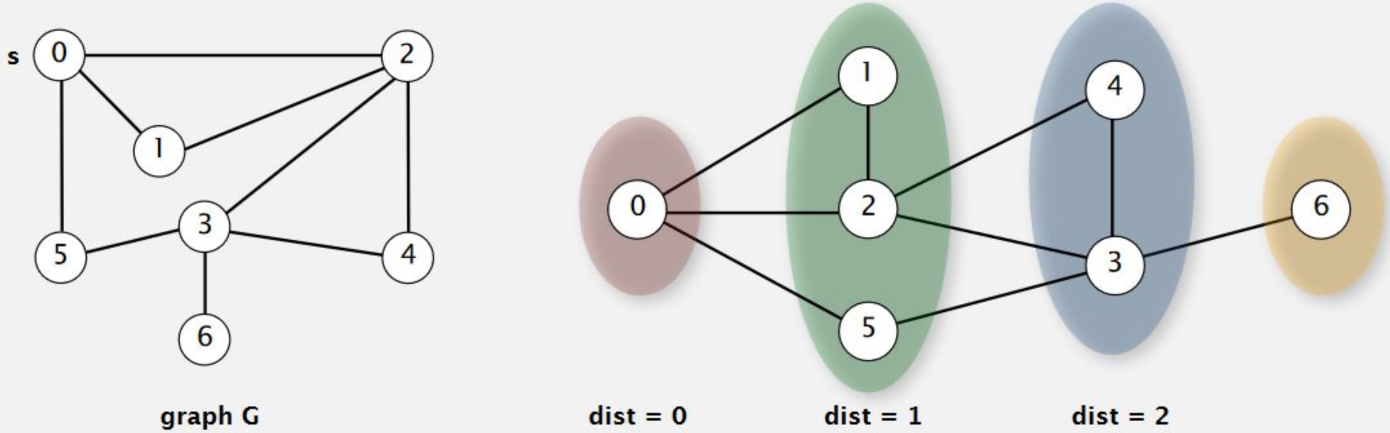
Breadth-first search properties

Q. In which order does BFS examine vertices?

A. Increasing distance (number of edges) from s.

queue always consists of ≥ 0 vertices of distance k from s, followed by ≥ 0 vertices of distance $k+1$

Proposition. In any connected graph G, BFS computes shortest paths from s to all other vertices in time proportional to $E + V$.



Решаване на задачи

https://www.hackerrank.com/contests/sda-hw-10

Топологична наредба на граф

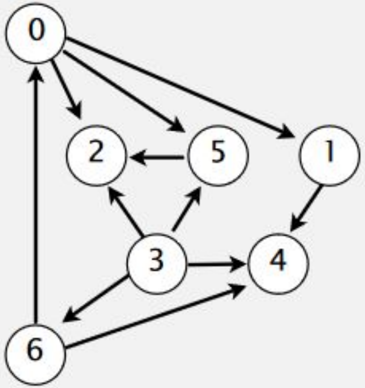
- Задача за **насочен ацикличен граф**

Топологична наредба:

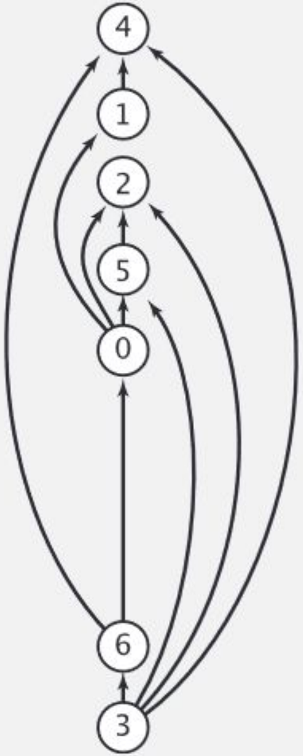
Да се наредят възлите на графа така, така че всяко насочено ребро да започва от възел по-напред в редицата и да отива във възел, който е по-назад в редицата.

0→5 0→2
0→1 3→6
3→5 3→4
5→2 6→4
6→0 3→2
1→4

directed edges



DAG



topological order

Алгоритми за решаване на Топологична наредба

- BFS
- DFS

Решение с BFS

```
L ← Empty list that will contain the sorted elements
S ← Set of all nodes with no incoming edge
while S is non-empty do
    remove a node n from S
    add n to tail of L
    for each node m with an edge e from n to m do
        remove edge e from the graph
        if m has no other incoming edges then
            insert m into S
if graph has edges then
    return error   (graph has at least one cycle)
else
    return L   (a topologically sorted order)
```

Решение с DFS

```
L ← Empty list that will contain the sorted nodes
while there are unmarked nodes do
    select an unmarked node n
    visit(n)

 function visit(node n)
    if n has a permanent mark then return
    if n has a temporary mark then stop   (not a DAG)
    mark n temporarily
    for each node m with an edge from n to m do
        visit(m)
    mark n permanently
    add n to head of L
```

graph problem	BFS	DFS	time
s-t path	✔	✔	$E + V$
shortest s-t path	✔		$E + V$
cycle	✔	✔	V
Euler cycle		✔	$E + V$
Hamilton cycle			$2^{1.657\,V}$
bipartiteness (odd cycle)	✔	✔	$E + V$
connected components	✔	✔	$E + V$
biconnected components		✔	$E + V$
planarity		✔	$E + V$
graph isomorphism			$2^{c \ln^3 V}$

Това е всичко за днес!

Какво следва:

- Следваща лекция - Най-къс път в граф, алгоритъм на Дейкстра
- Вторник(16.12) от 17:15 контролно с задачи от материала до сега (включващ и Хеш таблица и Граф)