CSE 30 5-11-21

Pa5: ext, 1 day (Friday)

difference between mul()
and rmul()

mul(self, n) implements self * n
rmul(self, n) " n * self

Graphs and Graph Algorithms

A <u>Graph</u> is a pair of sets

$$G = (V, E)$$

Vertex set $V(G)$ edge set $E(G)$

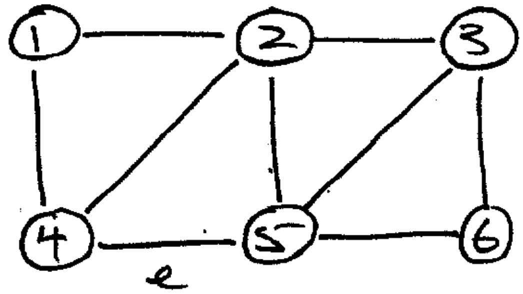
where

$$E \subseteq V^{(2)} = \{\text{2-element subsets of } V\}$$

$$= \{\text{unordered pairs of vertices}\}$$

Ex.

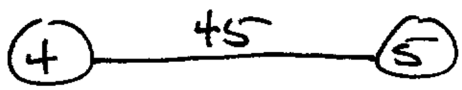
G



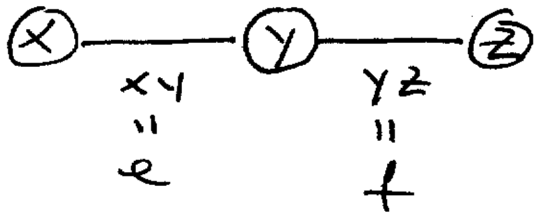
$V = \{1, 2, 3, 4, 5, 6\}$

$E = \{\{1,2\}, \{1,4\}, \{2,3\}, \{2,4\}, \{2,5\},$
$\{3,5\}, \{3,6\}, \{4,5\}, \{5,6\}\}$

other notation: $\{4,5\} = 45 = 54 = e$



Given picture:



we say that:

- x and y are <u>adjacent</u>
- e <u>joins</u> x to y
- e has <u>ends</u> x and y

- e and y are <u>incident</u>
- e and f are <u>adjacent</u>

<u>Also</u>: the <u>degree</u> of a vertex is the number of edges incident with it, so

- $\deg(x) = 1$, $\deg(y) = 2$, $\deg(z) = 1$.

Given a graph G and $x, y \in V(G)$, an <u>x-y walk</u> in G is a sequence

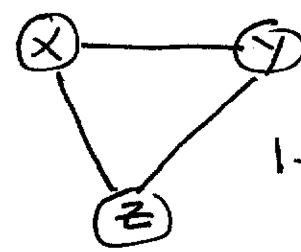
$$x = v_0, v_1, v_2, \text{--------}, v_k = y$$

of vertices in which each consecutive pair are adjacent: $\{v_{i-1}, v_i\} \in E(G)$ for $1 \le i \le k$.

we say that the <u>length</u> of a walk is the number of edge traversals, i.e. k.

A walk is called <u>closed</u> iff $x = y$

<u>Ex.</u> x, y, z, x



length = 3

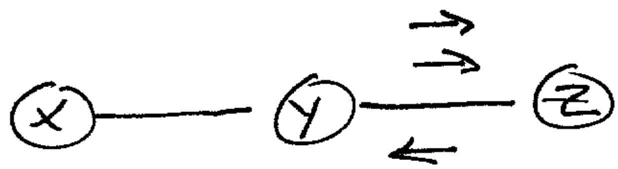
A walk is <u>trivial</u> iff length = 0

x

A **trail** is a walk in which no edge is repeated.

Ex a walk that is not a trail

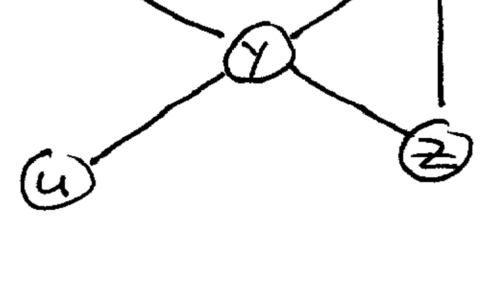
x, y, z, y, z



A **Path** is a trail in which no vertex is repeated. (exception: closed)

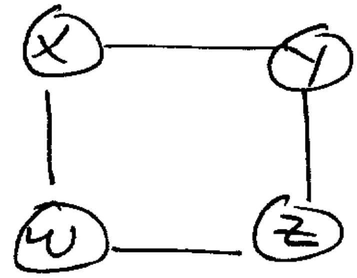
Ex a trail that is not a Path

x, y, z, w, y, u



A *cycle* is a non-trivial closed path

Ex.



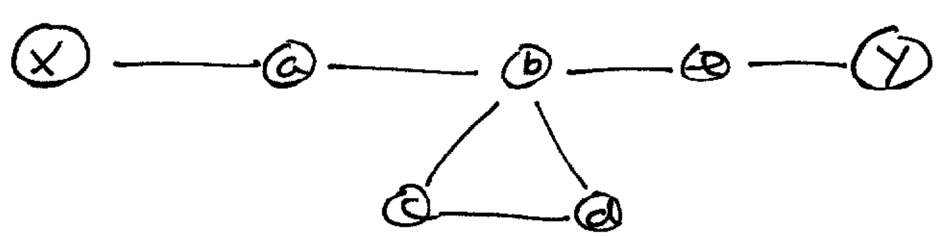
length = 4

x, y, z, w, x

Defn

Given $x, y \in V(G)$, we say that y is *reachable* from x iff G contains an $x-y$ path.

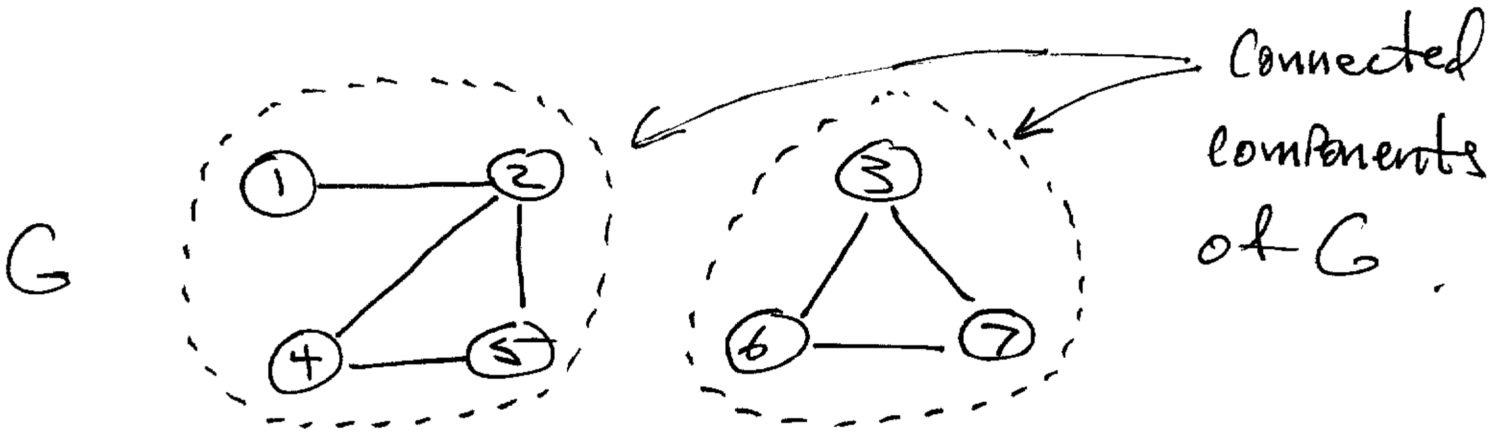
note: if G contains an $x-y$ walk, then it also contains an $x-y$ path



Defn

A Graph G is called connected iff for all $x, y \in V(G)$, y is reachable from x (and x from y).

Ex. a disconnected graph is one that is not connected



$V = \{1, 2, 3, 4, 5, 6, 7\}$

$E = \{12, 24, 25, 45, 36, 37, 67\}$

Defn

A graph H is called a subgraph of G iff

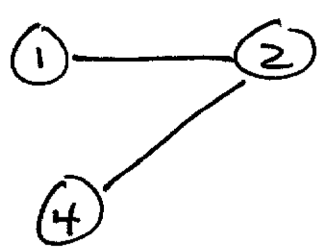
- $V(H) \subseteq V(G)$

and

- $E(H) \subseteq E(G)$

Ex. connected subgraph

$V = \{1, 2, 4\}$, $E = \{12, 24\}$



Ex disconnected subgraph

$V = \{4, 5, 3, 7\}$

$E = \{45, 37\}$

Ex. not a subgraph, since not a graph

$V = \{1, 2\}$ $E = \{45, 37\}$

Defn

A connected component of a graph G is a subgraph H satisfying both

(1) H is connected

and (2) H is maximal with respect to property (1). (i.e. no subgraph of G that has H as a (proper) subgraph is also connected.)

Problems:

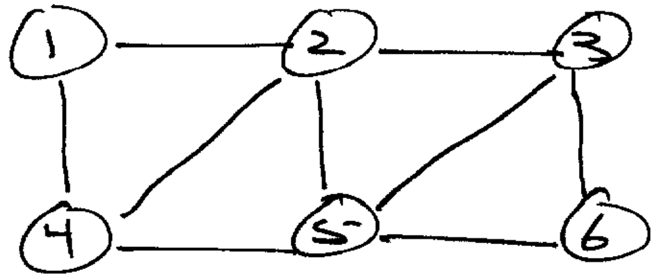
- Given G and $x \in V(G)$, determine the set of vertices reachable from x.
- Given G, determine the connected components of G.

How to represent a Graph in Python as a class?

Adjacency list representation:
maintain for each vertex $x \in V(G)$ a list of adjacent verticies

$\boxed{\text{adj}[x] : \text{---- neighbors of } x \text{ -----}}$

Ex.



1 : 2, 4

2 : 1, 3, 4, 5

3 : 2, 5, 6

4 : 1, 2, 5

5 : 2, 3, 4, 6

6 : 3, 5

Ex. see graph.py

EX