

-
- Quiz 4: next Tuesday
 - Pa5: ext. 2 days (to Sunday) last!
 - lab3: soon
 - Pa6: soon
 - Autograder for Pa5?

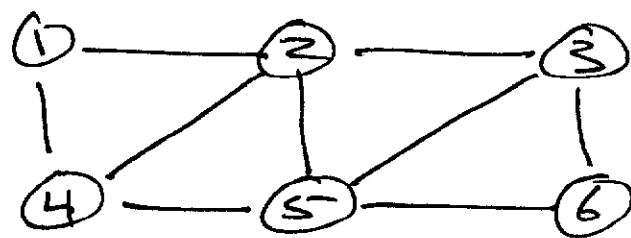
Graphs : Problems

- Given G and $s \in V(G)$ (called the source vertex) determine the set of vertices reachable from s .

- Given G , determine all connected components. Also Query any two vertices x, y .
Return True iff x is reachable from y (and y from x).

To solve these, use adjacency list representation.

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

see class Graph in module
graph.py

Exercise

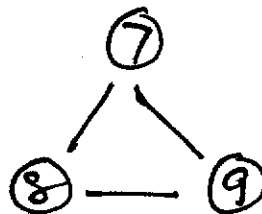
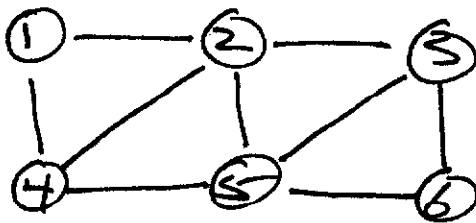
write

`def remove_vertex(self, x)`

`def remove_edge(self, e)`

Ex.

G



1	:	2, 4
2	:	1, 3, 4, 5
3	:	2, 5, 6
4	:	1, 2, 5
5	:	2, 3, 4, 6
6	:	3, 5
7	:	8, 9
8	:	7, 9
9	:	7, 8

Trace: $G.\text{reachable}(3)$ ↖ source

undiscovered = $\{1, 4, 5, 7, 8, 9\}$

discovered: $\{3, 2, 5, 6, 1, 4\}$

finished: $\{3, 2, 5, 6, 1, 4\}$

Exercise

• Trace: $G.\text{reachable}(9)$

• Trace: $G.\text{reachable}(6)$

To solve the 'find components'

problem, add a new member to

Graph class: dictionary of component labels. Keys are vertices, values are component labels (ints),

Trace : C. find components ()

completed = { 1, 2, 3, 4, 5, 6, 7, 8, 9 }

 ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

component[] : 1 1 1 1 1 1 2 2 2

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

component[] : 1 1 1 1 1 1 2 2 2

Problem:

find min length paths from a source vertex, to every vertex reachable from source

Soln Breadth First Search (BFS)