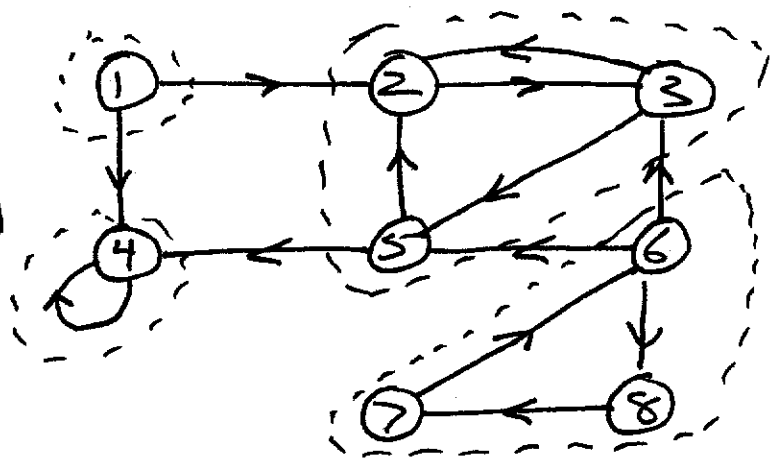
CSE 30 6-3-21

The graphs.py module:

Ex

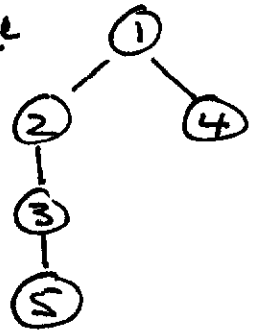
s = 1



	adj	dist.	Predecessor
1	2, 4	~~∞~~ 0	n
2	3	~~∞~~ 1	~~n~~ 1
3	2, 5	∞ 2	n 2
4	4	~~∞~~ 1	~~n~~ 1
5	2, 4	~~∞~~ 3	~~n~~ 3
6	3, 5, 8	∞	n
7	6	∞	n
8	7	∞	n

Q: ~~X~~ ~~Z~~ ~~4~~ ~~5~~ ~~8~~

Tree



Exercise

reachable(1) → {1, 2, 3, 4, 5}

Exercise (harder)

In DirectedGraph class, define functions

- findStrongComponents()
- getStrongComponent()
- sameStrongComponent()

add dictionary

_strong_component

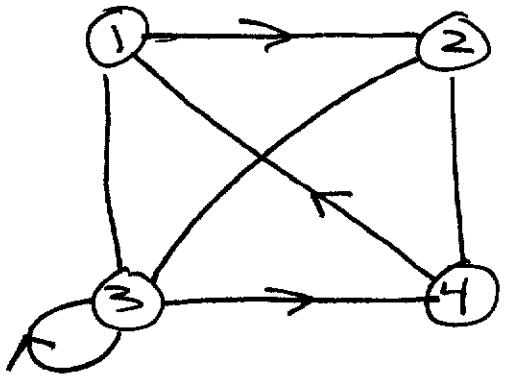
keys: $x \in V(G)$

values: _strong_component[x] =
int lable giving the
strong comp. of x.

Defn

A **General Graph** is a pair $G=(V,E)$ where E may consist of both directed and undirected edges.

Ex



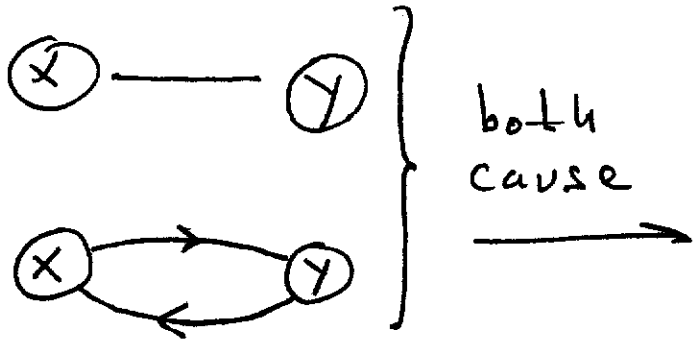
$V = [1, 2, 3, 4]$

$E = [\{1,3\}, \{2,3\}, \{2,4\}, (1,2), (4,1), (3,4), (3,3)]$

adj lists:

1 : [2, 3]
2 : [3, 4]
3 : [1, 2, 3, 4]
4 : [1, 2]

note:

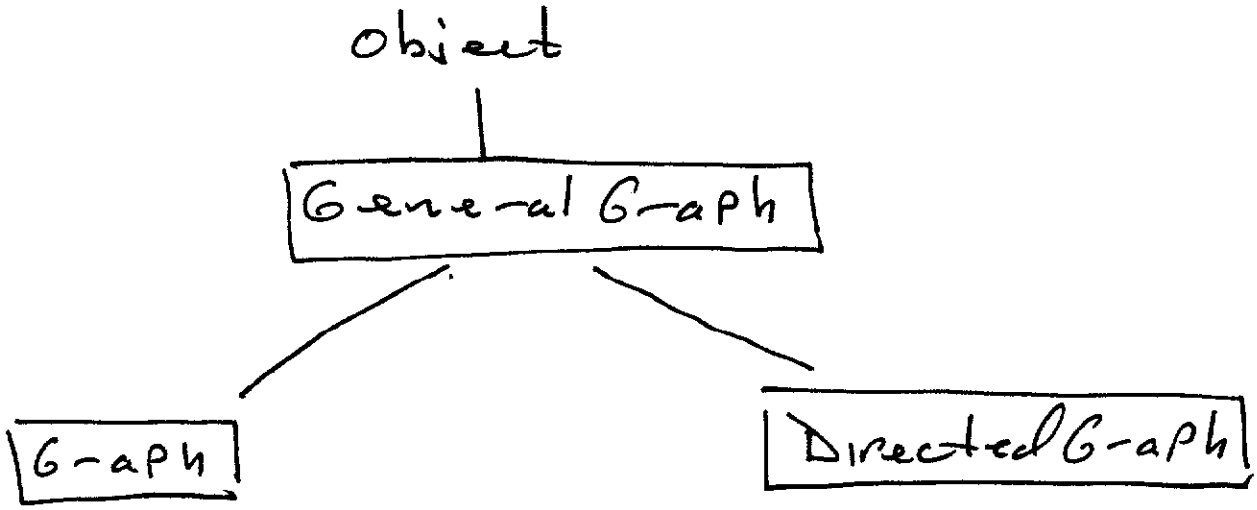


adj list

x : y

y : x

Exercise create 3 classes



Goal: build

findStrongComponents(self)

note

strong_component[x] $\subseteq$ reachable(x)

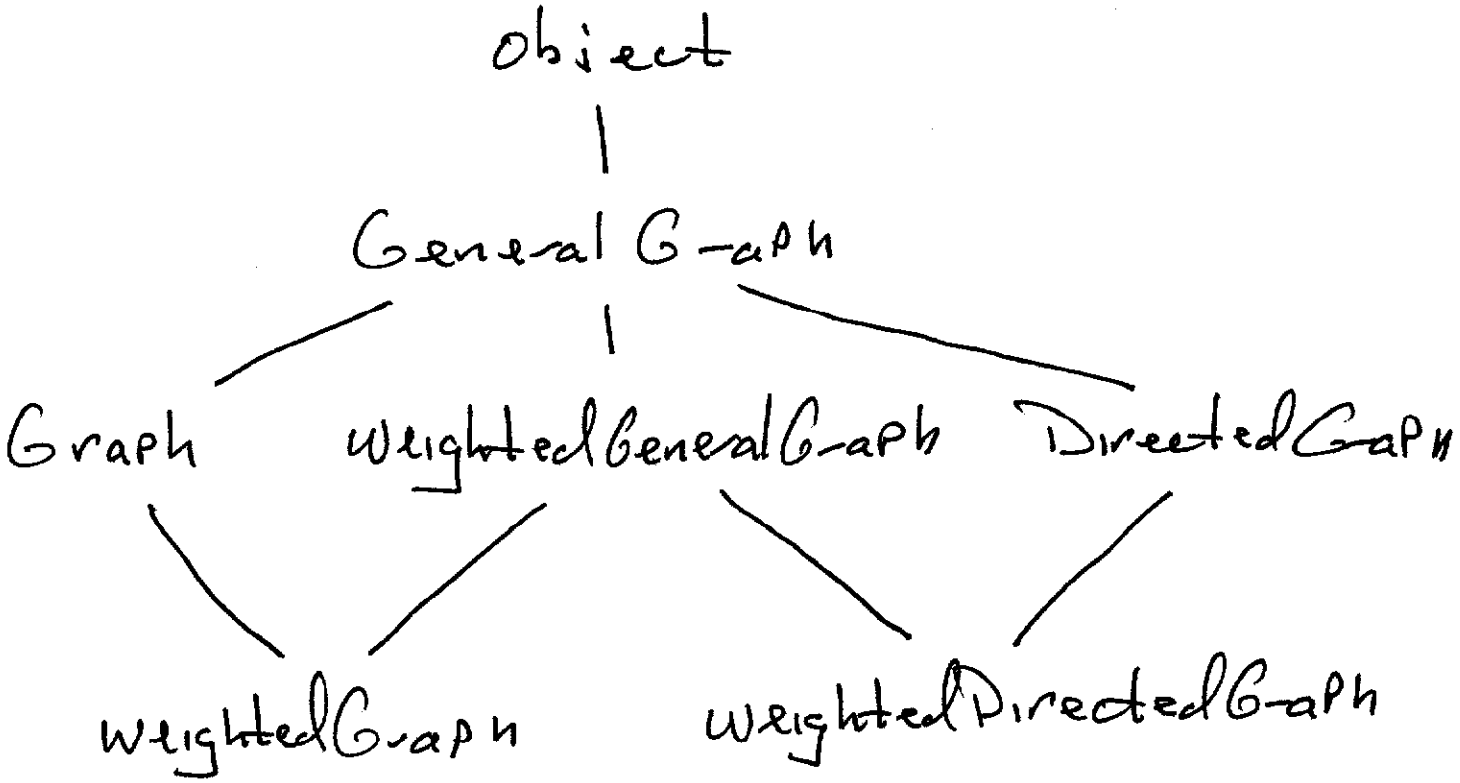
Further Project:

define weighted (General) Graph

$w : E \longrightarrow \mathbb{R}$

define weightedGeneralGraph

Create:


object
GeneralGraph
Graph
WeightedGeneralGraph
DirectedGraph
WeightedGraph
WeightedDirectedGraph