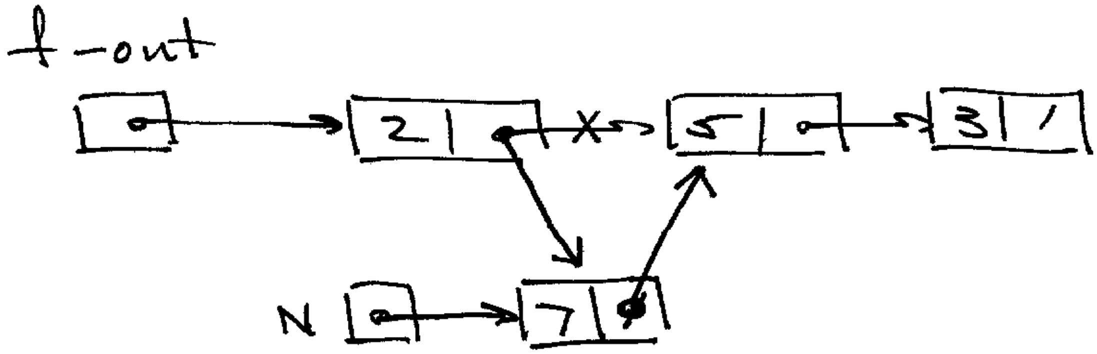
CSE 30 5-20-21

quiz4: solutions posted

lab3: ext. 1 day, to Sat,

Pa6:

Problem 2 on quiz4:



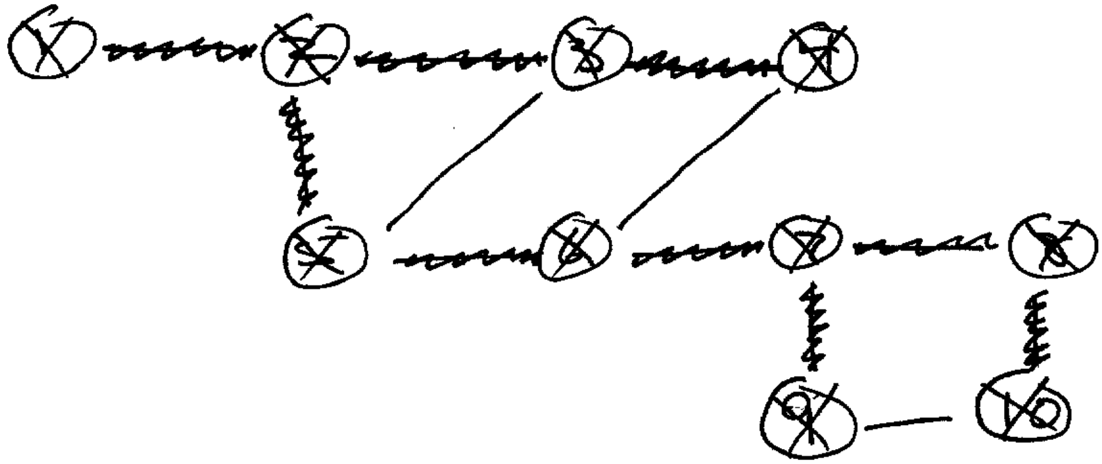
N = Node(7)

N.next = front.next

front.next = N

Another BFS() example

source = 2

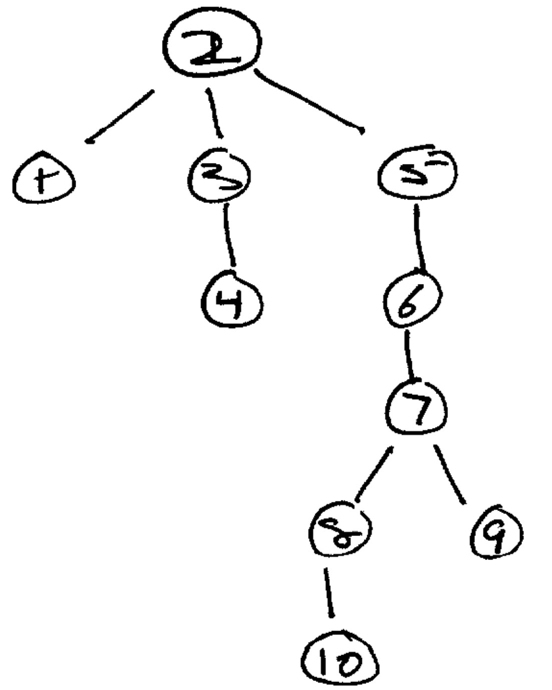


undiscovered : ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~

discovered : ~~2~~ ~~1~~ ~~3~~ ~~5~~ ~~4~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~

finished : $\underbrace{2}_{0}\ \underbrace{1\ 3\ 5}_{1}\ \underbrace{4\ 6}_{2}\ \underbrace{7}_{3}\ \underbrace{8\ 9}_{4}\ \underbrace{10}_{5}$

predecessor subgraph



dist

0

1

2

3

4

5

Pa6: Graph coloring

Defn

A graph G is called <u>k-colorable</u> iff it is possible to assign one of a set of k colors to each vertex $x \in V(G)$ so that no two adjacent vertices have the same color. i.e.,

$$(*) \qquad \{x, y\} \in E(G) \implies \text{color}[x] \neq \text{color}[y]$$

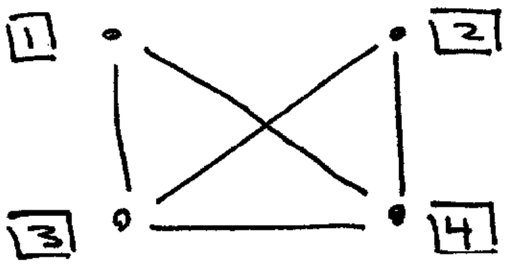
Such an assignment is called a <u>k-Coloring</u> or <u>Proper k-coloring</u>

color: $\alpha \quad \neq \quad \beta$

Usually take color set to be

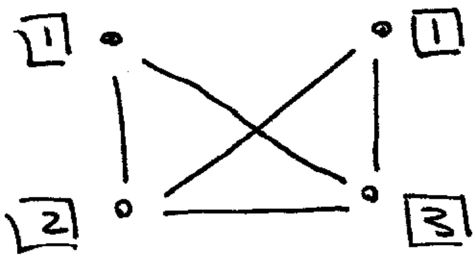
$$\{1, 2, 3, \ldots, k\}$$

Ex.



4-coloring

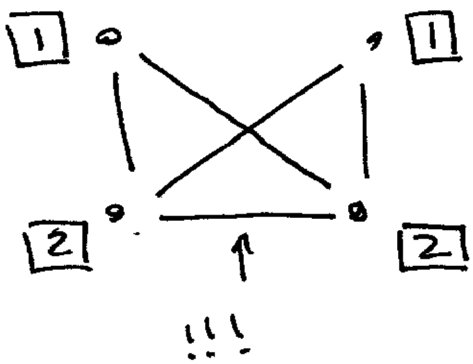
Ex.



3-coloring

3-chromatic

Ex.



not a proper 2-coloring

Defn

If G is k-colorable, but not $(k-1)$-colorable, we say G is <u>k-chromatic</u>, and we call k its <u>chromatic number</u>.

Notation: $\chi(G) = k$

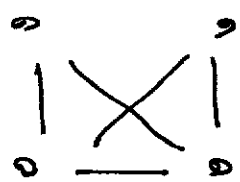
Theorem

if G is a graph with $n = |V(G)|$,

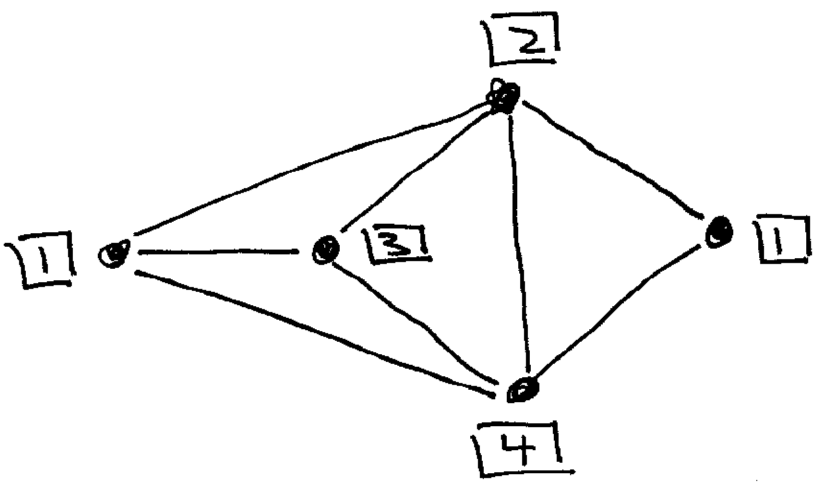
then $\chi(G) \leq n$.

why? give each vertex a unique color from $\{1, 2, \ldots, n\}$

Ex.

G  $\chi(G) = 3$

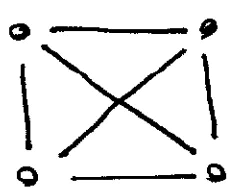
Ex.

G 

$\chi(G) = 4$

Defn

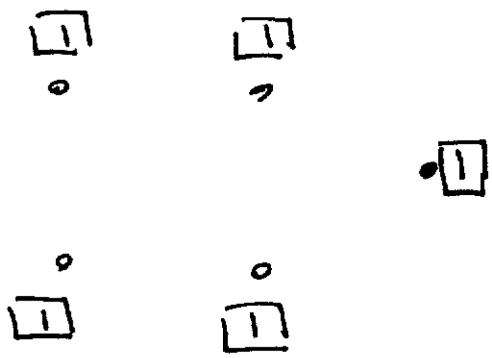
A graph is called **complete** iff for all $x, y \in V(G)$ there is an edge $\{x, y\} \in E(G)$, i.e., every pair of vertices are joined by an edge.

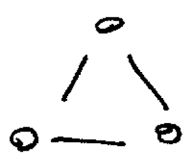
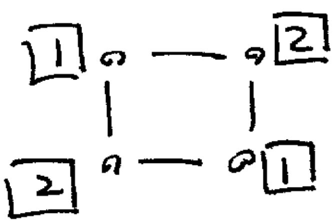
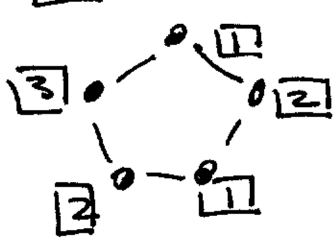
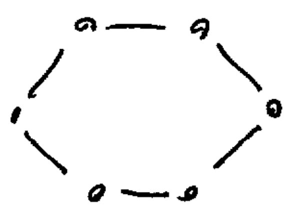
Complete Graphs		χ
K_1		1
K_2		2
K_3		3
K_4		4
K_5		5

Theorem: $\chi(K_n) = n$

Theorem: $E(G) = \phi \Rightarrow \chi(G) = 1$

Ex G:



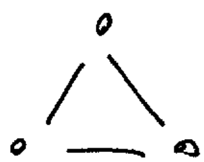
Cycles		χ
C_3		3
C_4		2
C_5		3
C_6		2
$\vdots$		

not allowed by defn

$\{x, x\} = \{x\}$

$\{x, y\} = \{x, y\}$

trivial



smallest cycle

Theorem

$$\chi(C_n) = \begin{cases} 2 & \text{if } n \text{ even} \\ 3 & \text{if } n \text{ odd} \end{cases}$$

Problem

Given a graph G with $|V(G)| = n$

(1) Determine a coloring of G with colors in $\{1, 2, \dots, n\}$

Further:

(2) Do it with the fewest possible colors: $\{1, 2, \dots, \chi(G)\}$

As we've seen, (1) is easy, while (2) is in general more difficult. Instead, we'll be hapy with

(3) <u>try</u> to do it with the fewest colors: $\{1, 2, \dots, k\}$ where

$$\chi(G) \leq k \leq n$$

strategy:

add attribute to Graph class

- color[x] : dictionary keys $\in V(G)$

  Values : $\{0, \underbrace{1, 2, \ldots, n}_{\text{Palette of Possible colors}}\}$ $n = |V(G)|$.

  0 ↑ uncolored

- ecs[x] : dictionary keys $\in V(G)$

  Values are sets, called <u>excluded color set</u>.

  ecs[x] = {those colors already assigned to the neighbors of x : adj[x] }

Then alter reachable() (or BFS()) in some way to assign colors properly.

Fcns add:

- def Color(self): ← return the set of used colors
- def getColor(self, x):