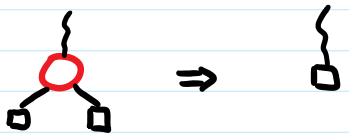
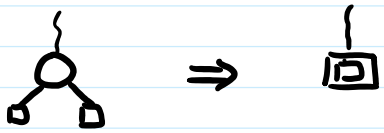
Red-Black Tree Delete

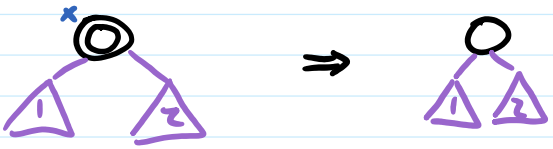
Do normal BST delete; if deleted node has 2 non-leaf children
then moved node takes deleted's color
do 1 non-leaf delete where moved node was





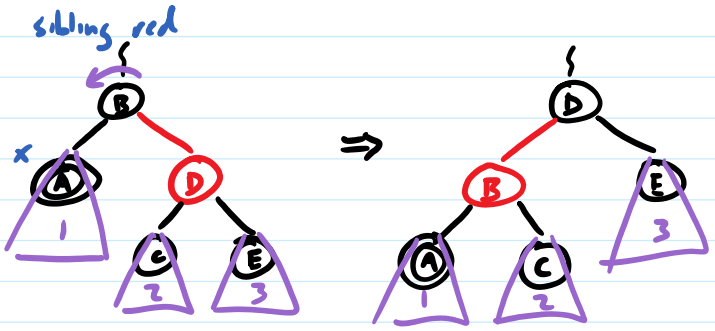
while x is doubly black

if

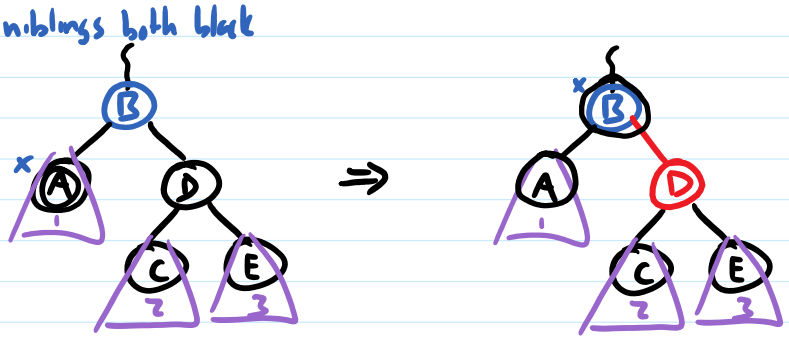


DONE

if



if

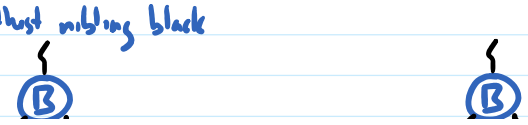


O + O = O

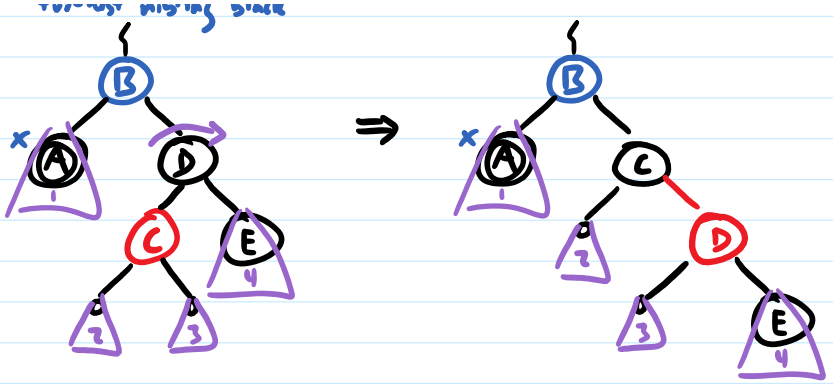
O + O = O

(keep going w/ x up one level)

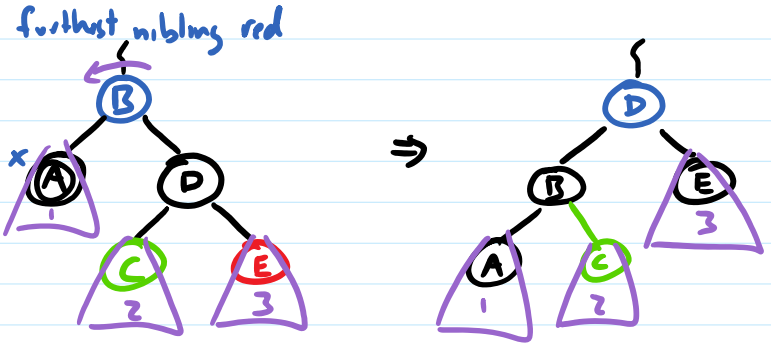
else if



else if

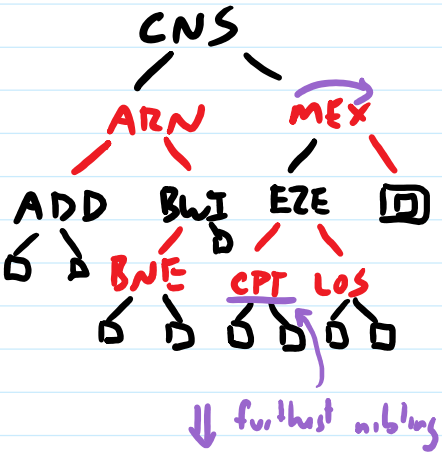


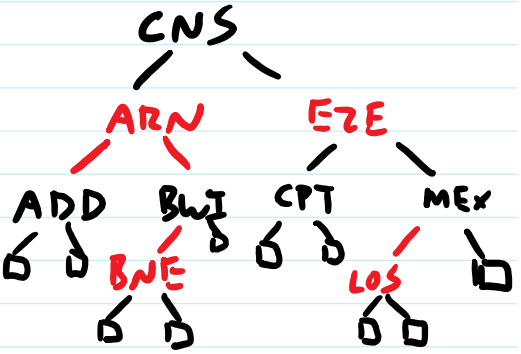
if furthest nibling red

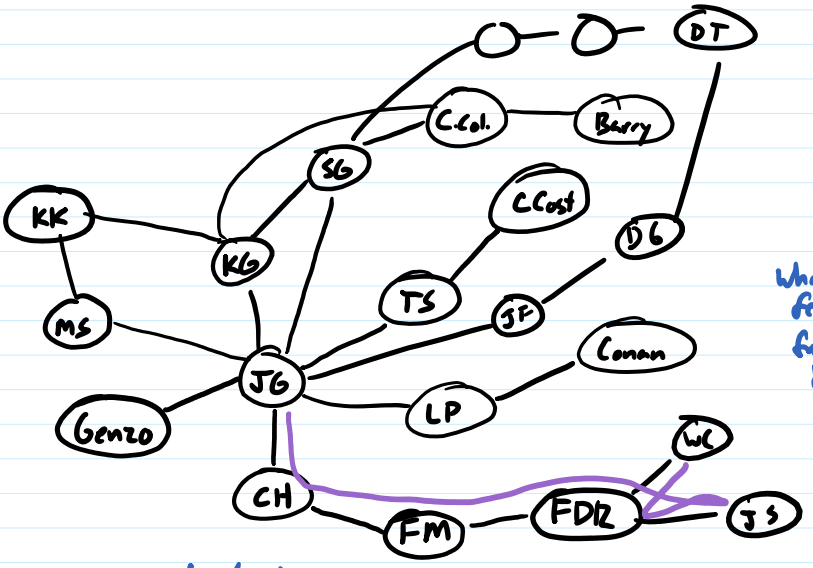


DONE


CNS
ARN
MEX
ADD
BWI
EZE
PVG
BNE
CPT
LOS
double-black leaf


CNS
ARN
MEX
ADD
BWI
EZE
BNE
CPT
LOS
furthest sibling node


CNS
ARN
EZE
ADD
BWI
CPT
MEX
BNE
LOS



what path has fewest edges from one person to another?

Graph: represents things and relationships between them
nodes/vertices — edges
people — relationships

path: seq of verts s.t. adj verts have edge in graph
JG, CH, FM, FDR, JS, FDR, WC

simple path: path w/no repeats JG, CH, FM, FDR, JS

cycle: cycle starting/ending at same place JG, KG, KK, MS, JG, KG, SG, JG

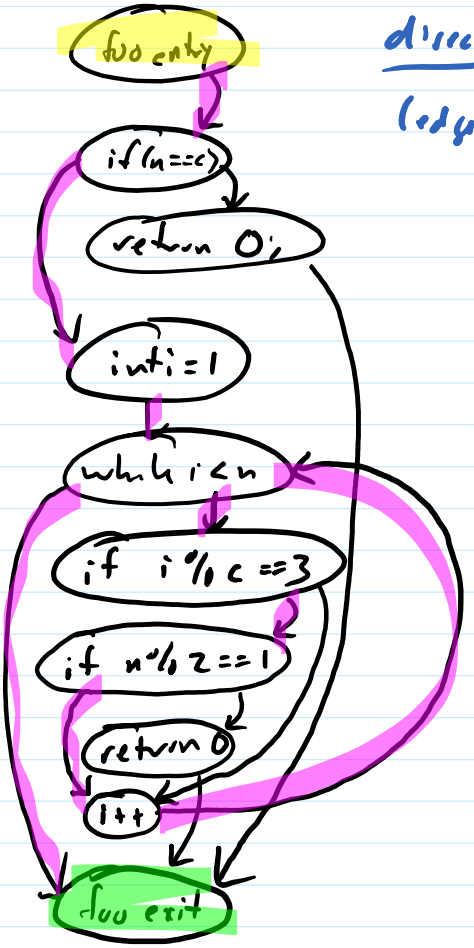
simple cycle: cycle w/ no repeats except start, end JG, KG, SG, JG

```
int foo(int n, int c)
{
    if (n == c)
    {
        return 0;
    }
    int i = 1;
    while (i < n)
    {
        if (i % c == 3)
        {
            if (n % 2 == 1)
            {
                return 0;
            }
        }
        i++;
    }
}
```

vertices: lines of code

edges: control flow

is there a path entry → exit that doesn't go through a return?



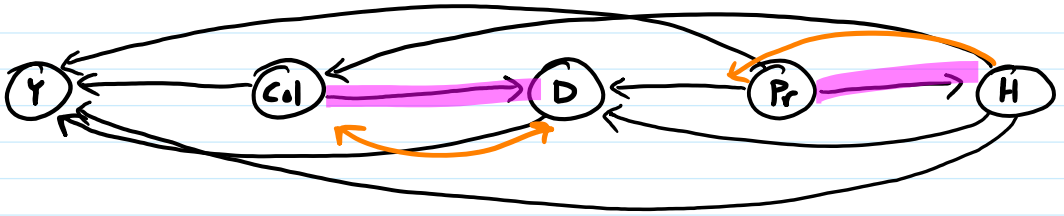
directed graph (edges have direction)

vertices: lines of code

edges: control flow

```
while (x>1)
    x--
foo()
```

verts: teams edge $u \to v$: u lost to v at least once

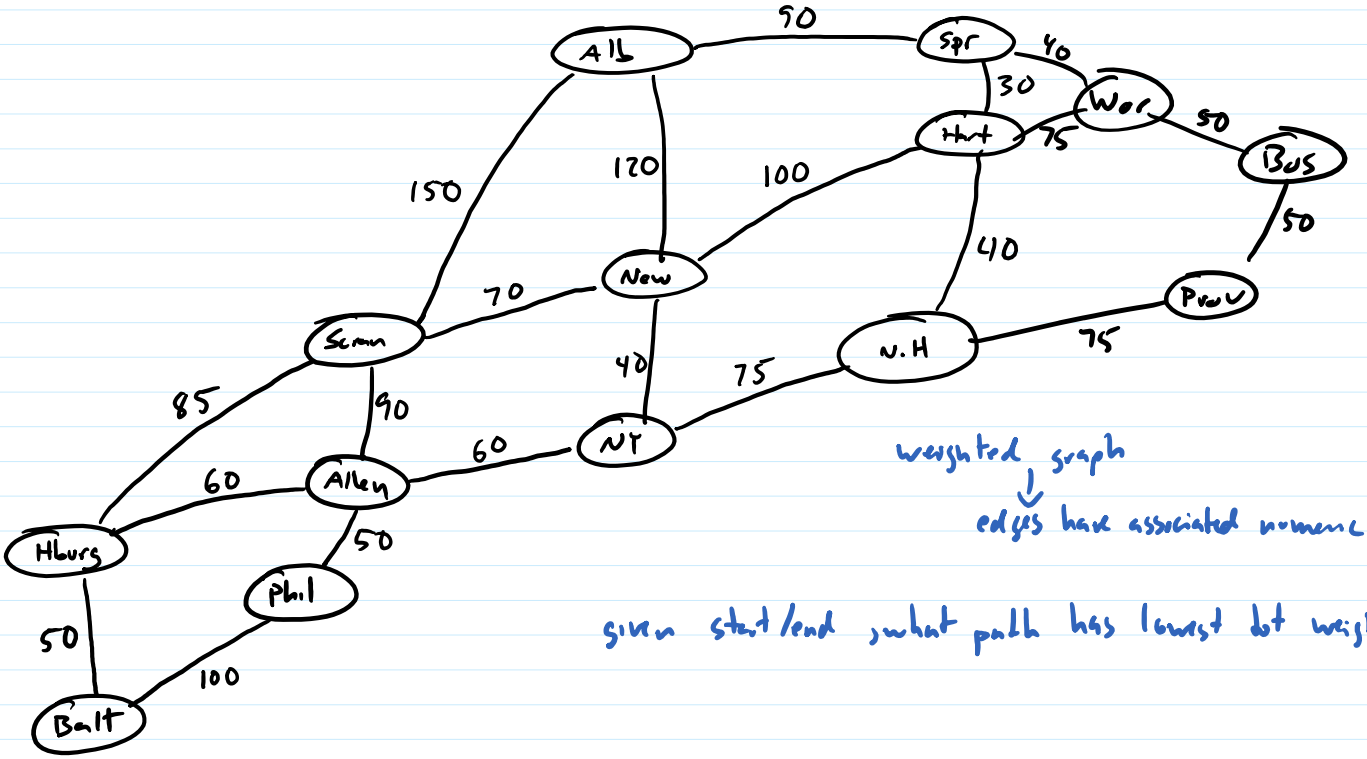


what ordering of verts minimizes edges in "wrong" direction

vertices

edges

FEEDBACK-ARC-SET
(NP-complete)

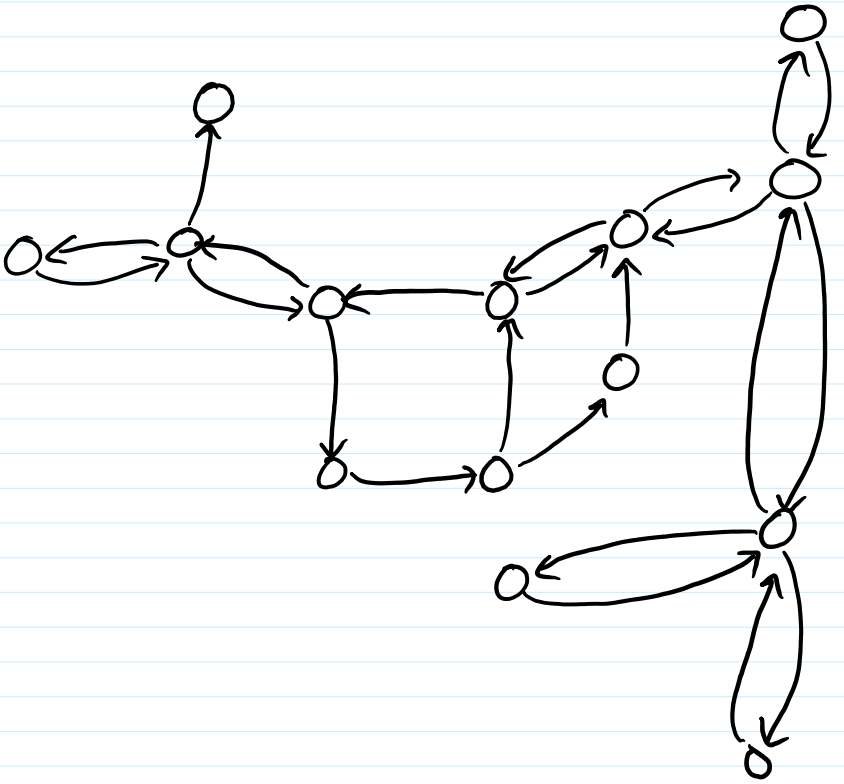


weighted graph

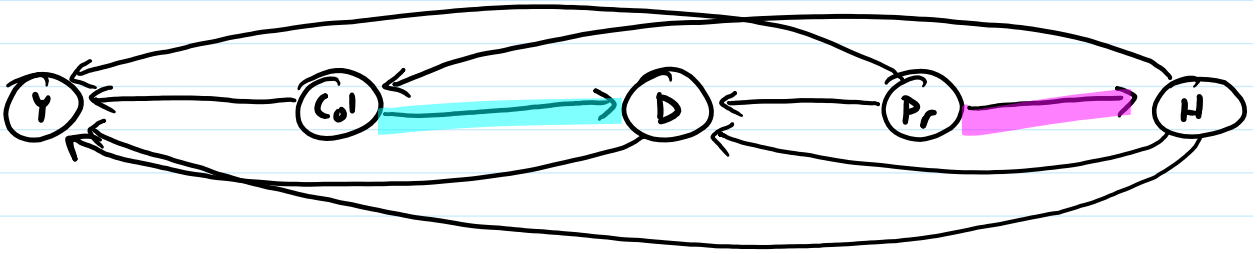
↓

edges have associated numeric weight

given start/end, what path has lowest tot weight

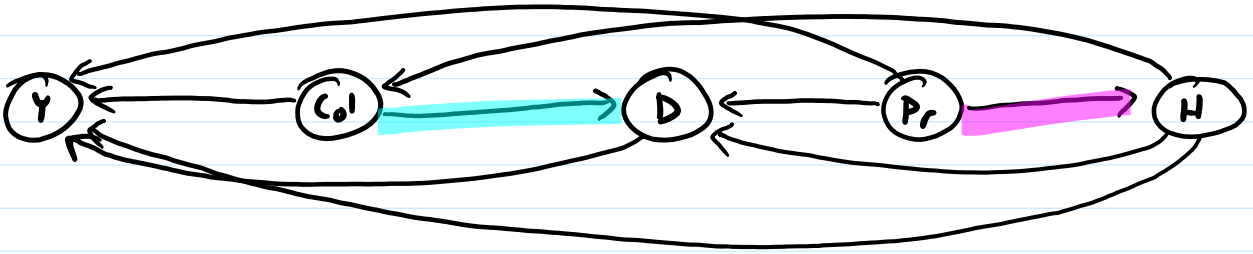


Graph Representation



Adjacency Matrix

from \ to	Y	Col	D	Pr	H
Y	F	F	F	F	F
Col	T	F	T	F	F
D	T	F	F	F	F
Pr	T	T	T	F	T
H	T	T	T	F	F



Adjacency List

Y :

Col : Y D

D : Y

Pr : ۲ D H

H : ۲ Col D