



# Compilers

---

## Graph Coloring

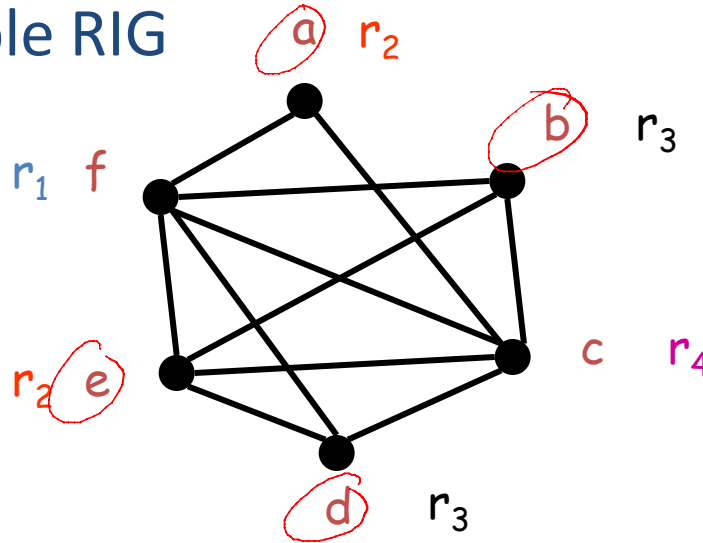
- A coloring of a graph is an assignment of colors to nodes, such that nodes connected by an edge have different colors



- A graph is k-colorable if it has a coloring with k colors

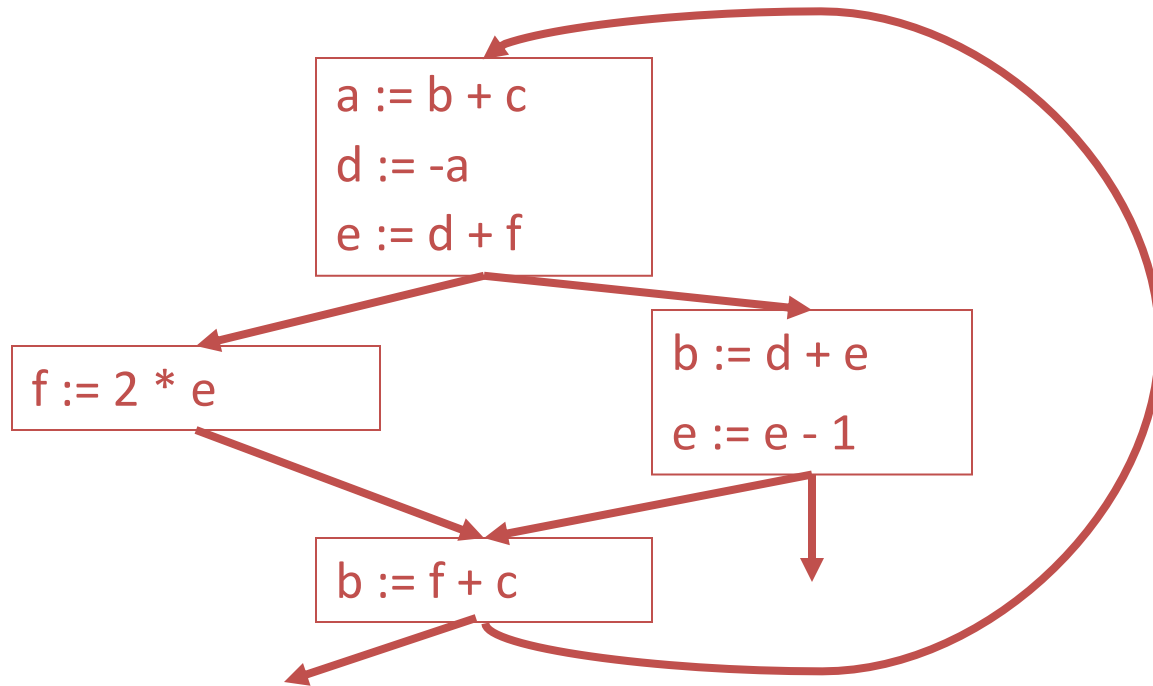
- In our problem, colors = registers
  - We need to assign colors (registers) to graph nodes (temporaries)
- Let  $k$  = number of machine registers
- If the RIG is  $k$ -colorable then there is a register assignment that uses no more than  $k$  registers

- Consider the example RIG

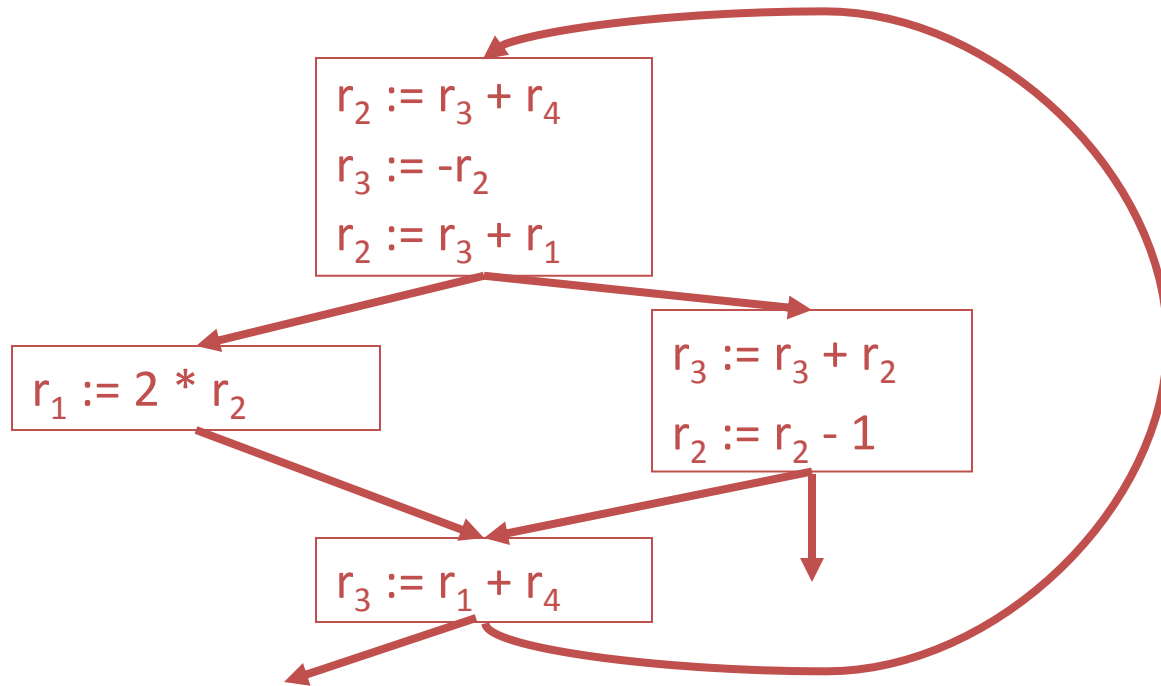


- There is no coloring with less than 4 colors
- There are 4-colorings of this graph

# Graph Coloring

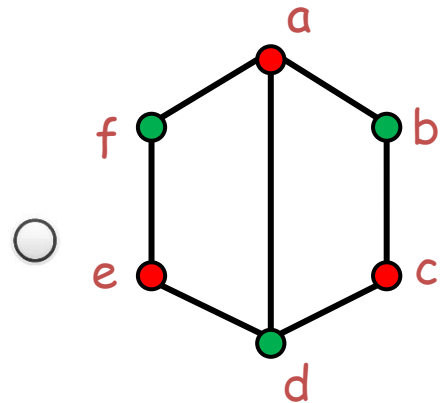
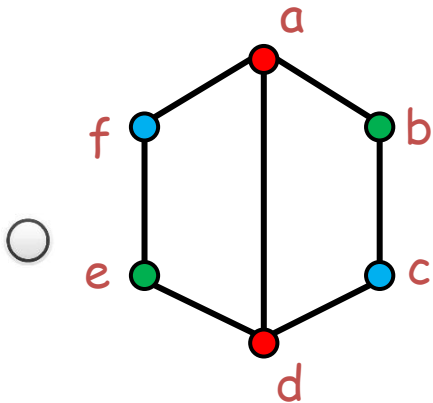
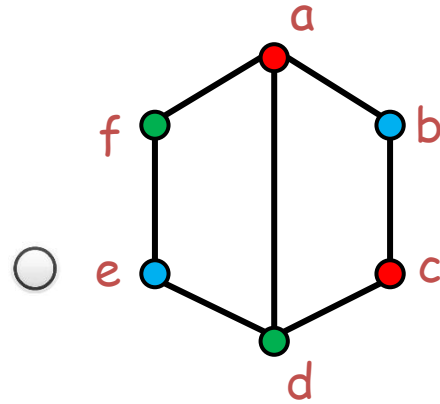
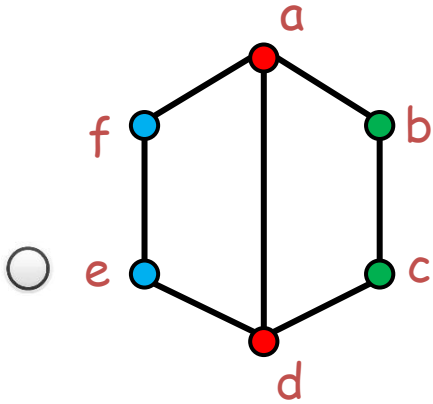


# Graph Coloring



Which of the following colorings is a valid minimal coloring of the given RIG?

## Graph Coloring



- How do we compute graph colorings?
- It isn't easy:
  1. This problem is very hard (NP-hard). No efficient algorithms are known.
    - *Solution: use heuristics*
  2. A coloring might not exist for a given number of registers
    - *Solution: later*



- Observation:
  - Pick a node t with fewer than k neighbors in RIG
  - Eliminate t and its edges from RIG
  - If resulting graph is k-colorable, then so is the original graph



- Why?
  - Let  $c_1, \dots, c_n$  be the colors assigned to the neighbors of t in the reduced graph
  - Since  $n < k$  we can pick some color for t that is different from those of its neighbors

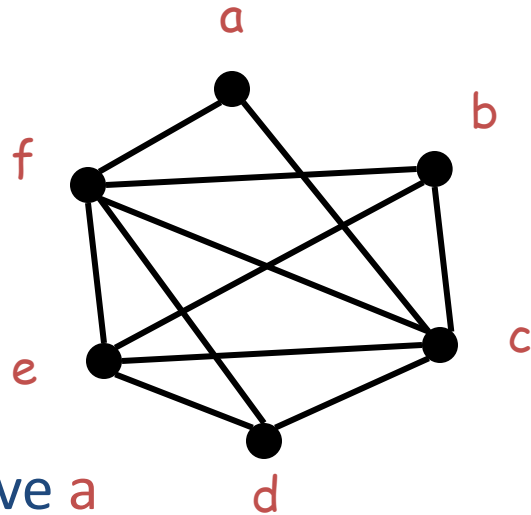
- 1
- The following works well in practice:

- Pick a node t with fewer than k neighbors
- Put t on a stack and remove it from the RIG
- Repeat until the graph is empty

- 2
- Assign colors to nodes on the stack

- Start with the last node added
- At each step pick a color different from those assigned to already colored neighbors

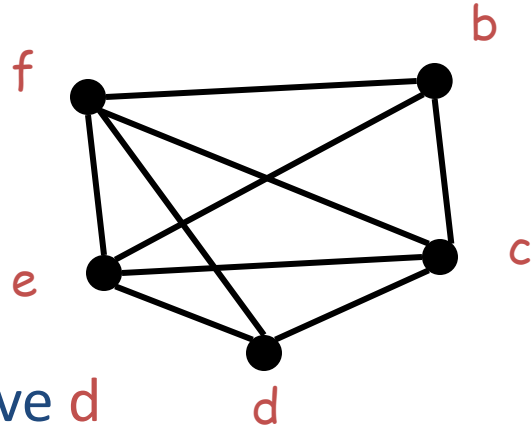
- Start with the RIG and with  $k = 4$ :



Stack: {}

- Remove a

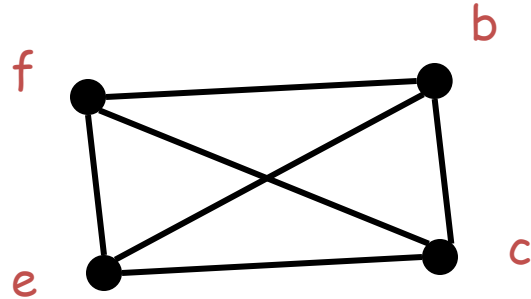
# Graph Coloring



Stack: {a}

- Remove  $d$

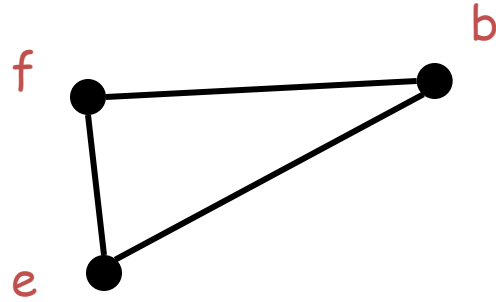
- Note: all nodes now have fewer than 4 neighbors



Stack: {d, a}

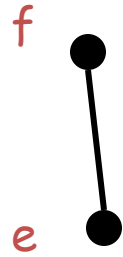
- Remove c

# Graph Coloring



Stack: {*c*, *d*, *a*}

- Remove *b*



Stack: {b, c, d, a}

- Remove e

*f* ●

Stack: {*e*, *b*, *c*, *d*, *a*}

- Remove *f*



Stack: {f, e, b, c, d, a}

- Empty graph – done with the first part!

- Now start assigning colors to nodes, starting with the top of the stack

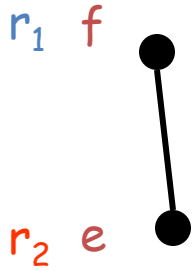
Stack: {f, e, b, c, d, a}

# Graph Coloring

$r_1$   $f$  ●

Stack: { $e$ ,  $b$ ,  $c$ ,  $d$ ,  $a$ }

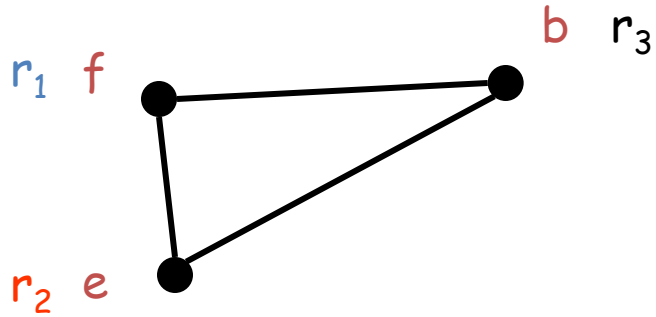
# Graph Coloring



Stack: { $b$ ,  $c$ ,  $d$ ,  $a$ }

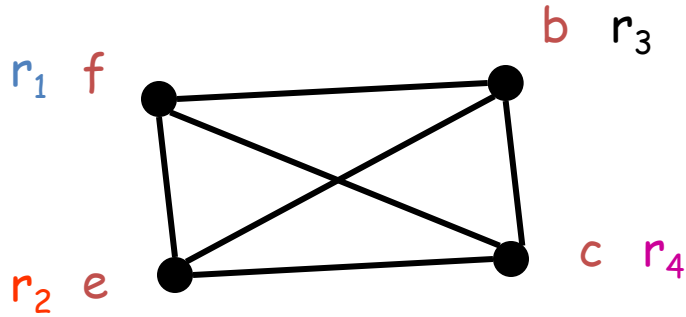
- $e$  must be in a different register from  $f$

# Graph Coloring



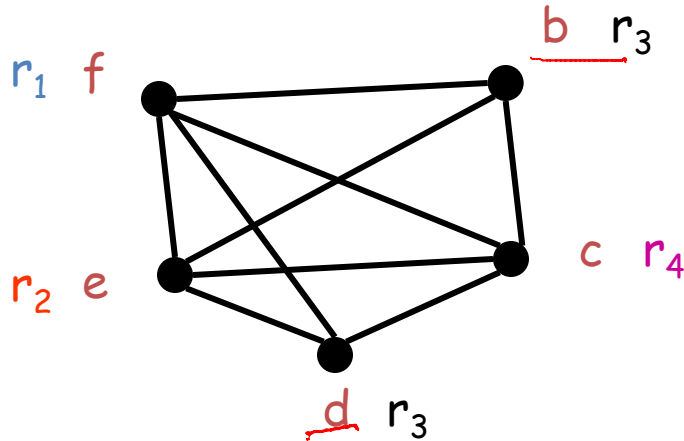
Stack: { $c$ ,  $d$ ,  $a$ }

# Graph Coloring



Stack:  $\{d, a\}$

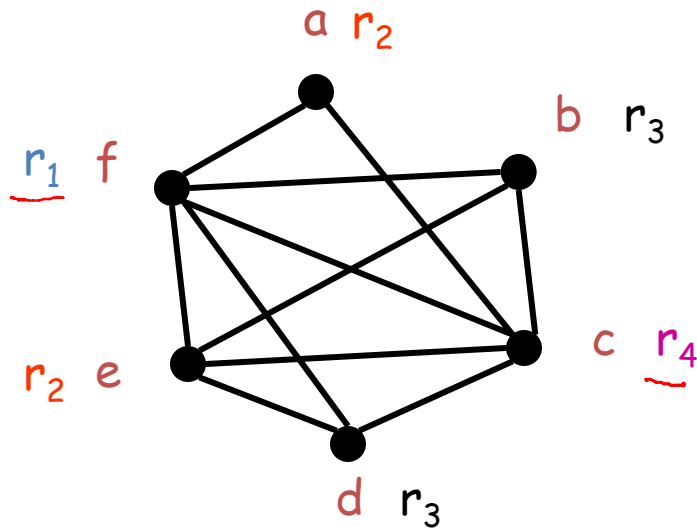
# Graph Coloring



Stack: { $\underline{a}$ }

- $d$  can be in the same register as  $b$

# Graph Coloring



Stack: {}



## Graph Coloring

For the given RIG and  $k = 3$ , which of the following are valid deletion orders for the nodes of the RIG?

- ☐ {d, e, c, b, a, f}
- ☐ {e, f, a, b, c, d}
- ☐ {d, c, b, a, f, e}
- ☐ {d, e, b, c, a, f}

