Register Allocation

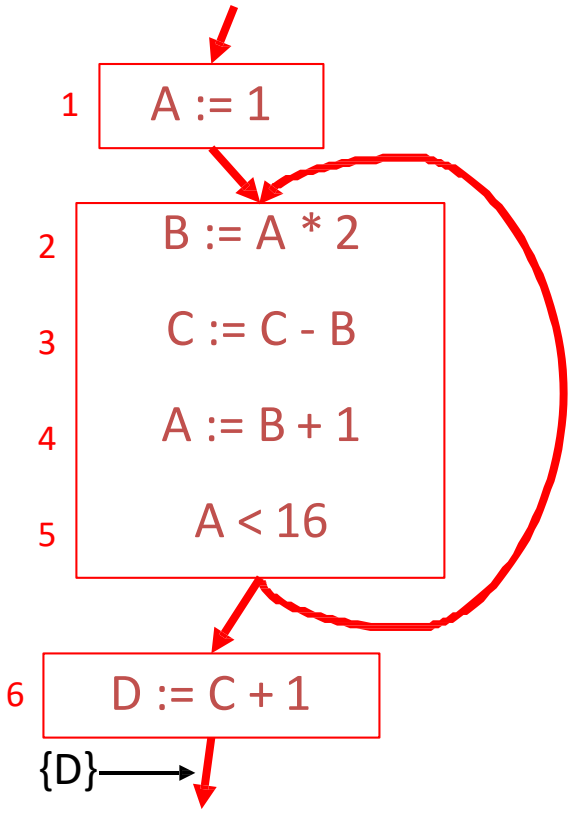
Lecture 13

Exercise

Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

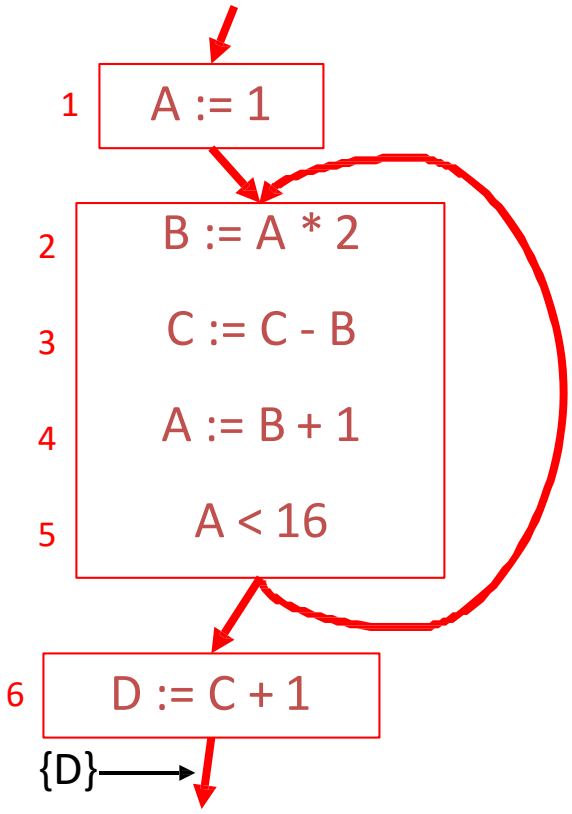
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

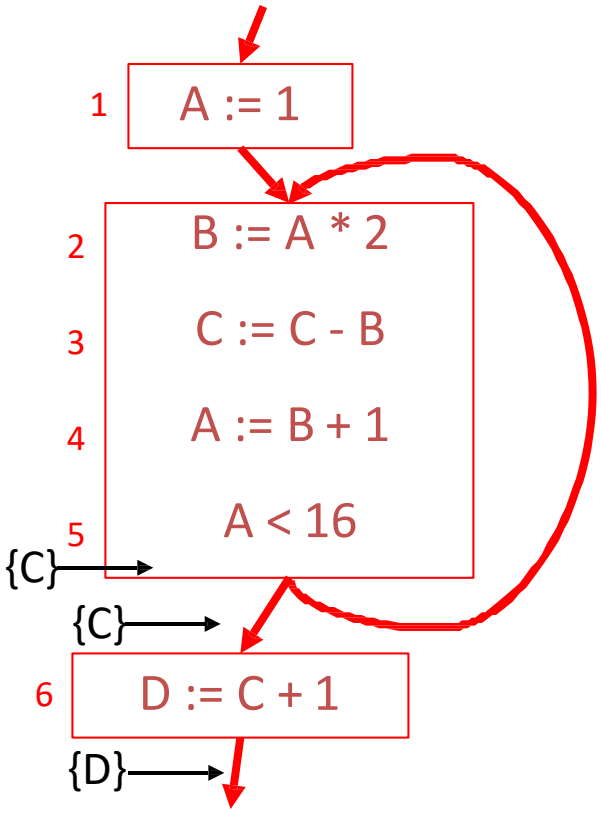
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

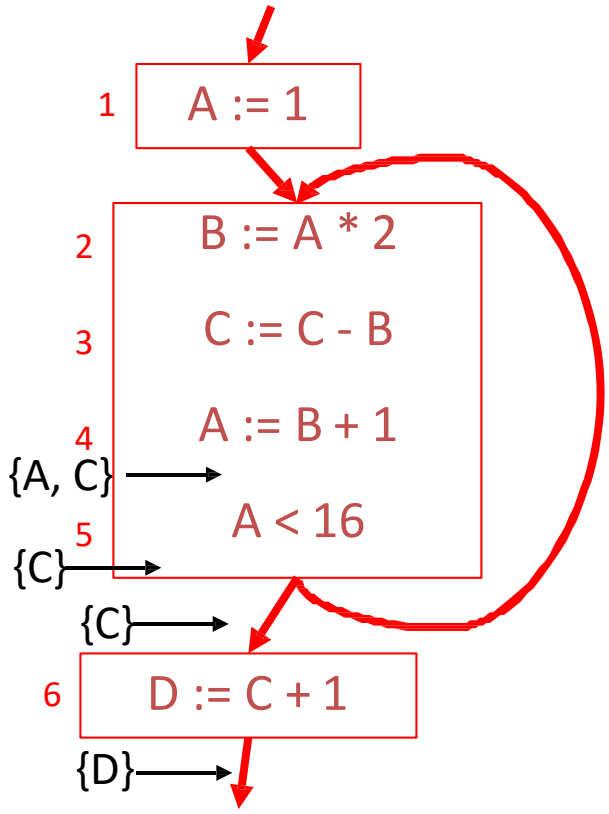
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

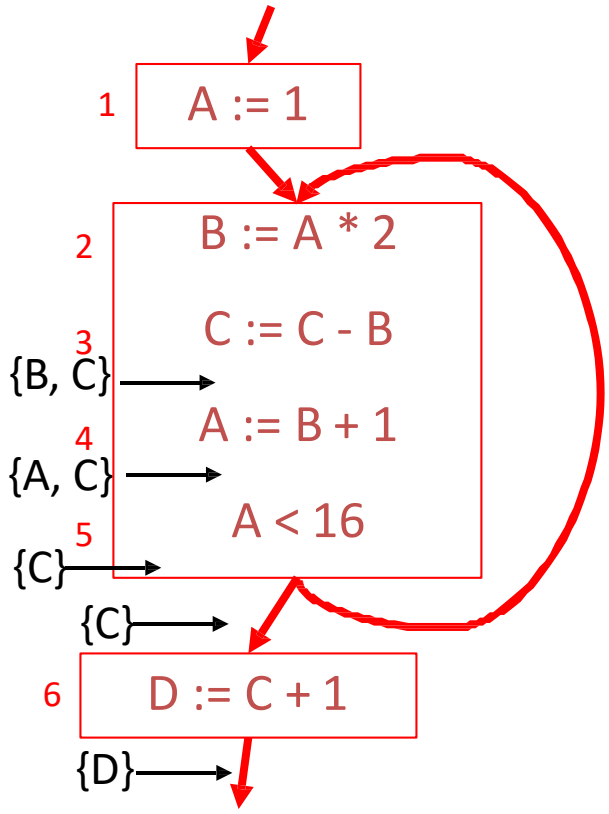
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

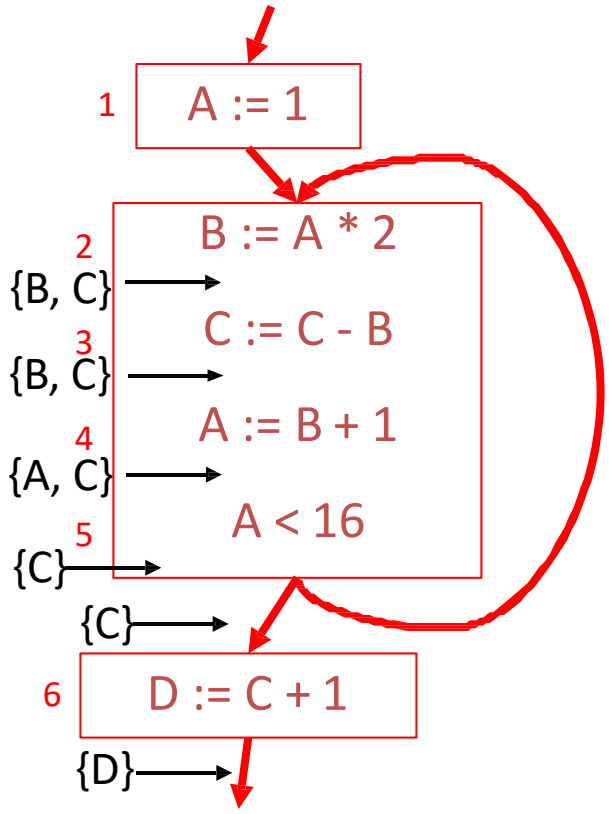
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

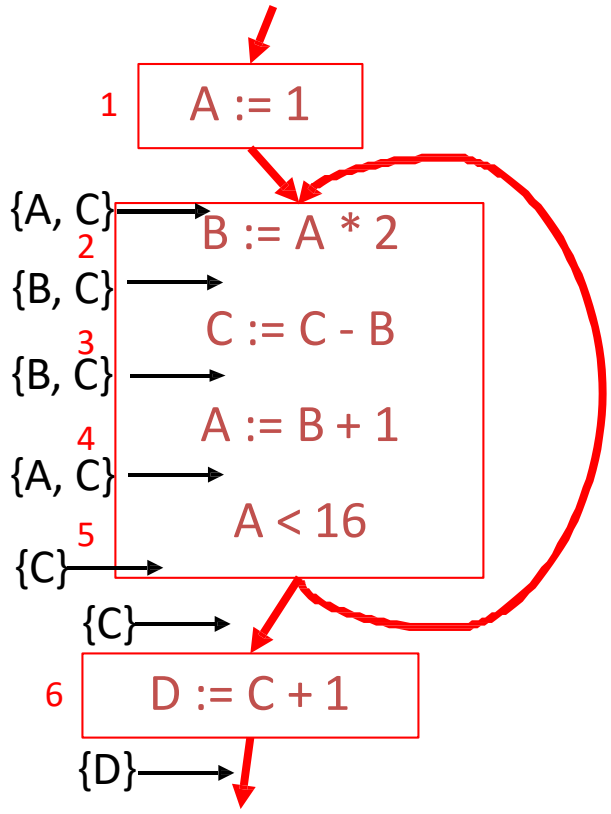
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

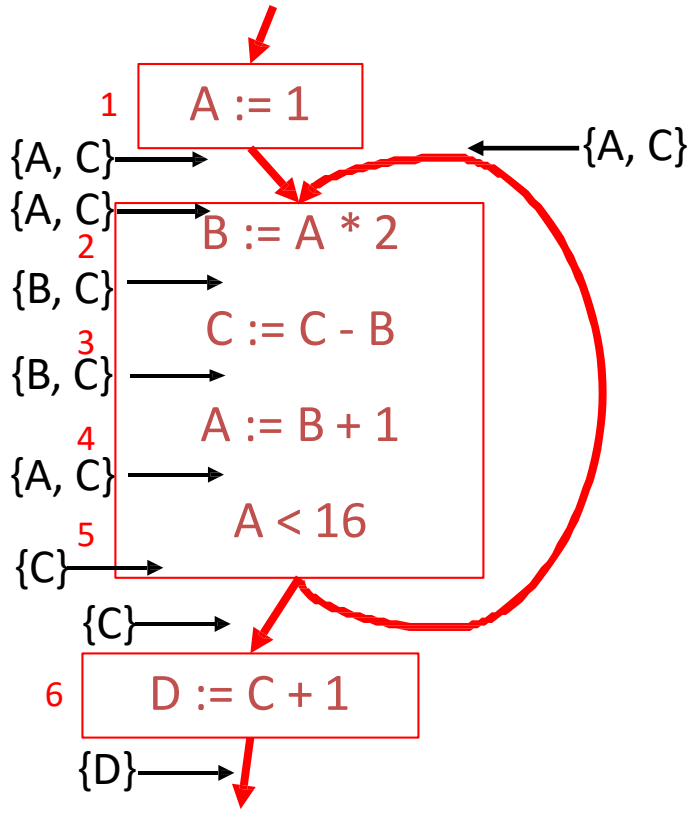
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

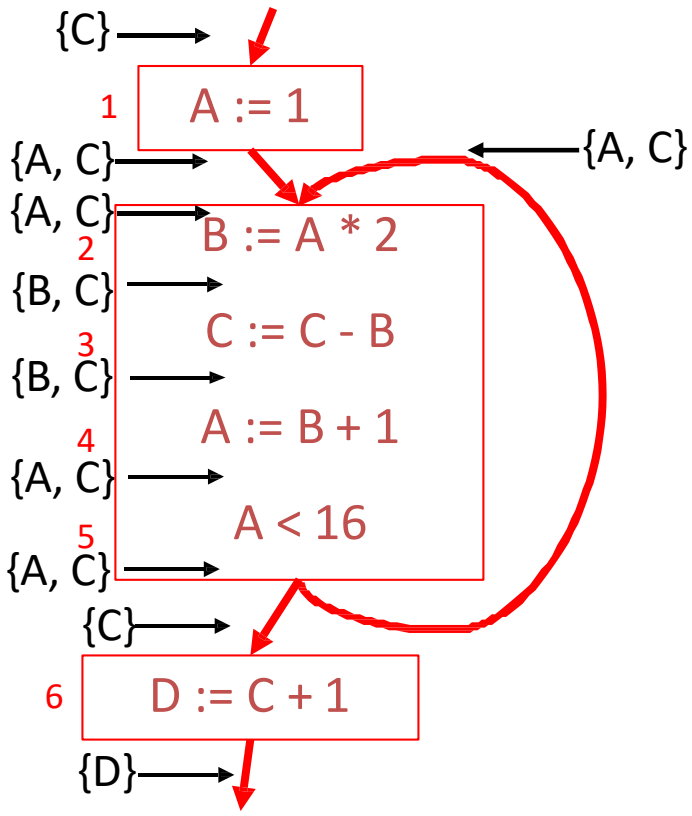
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Question?

Which of the following pairs of temporaries interfere in the code fragment given at right?

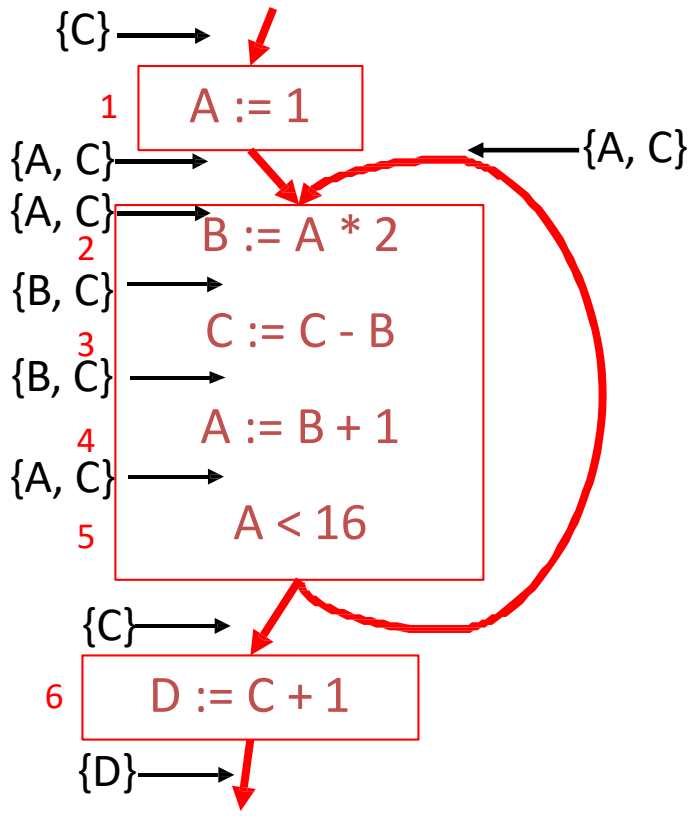
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D



Answer!

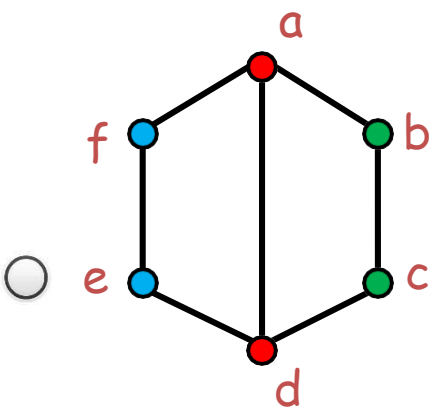
Which of the following pairs of temporaries interfere in the code fragment given at right?

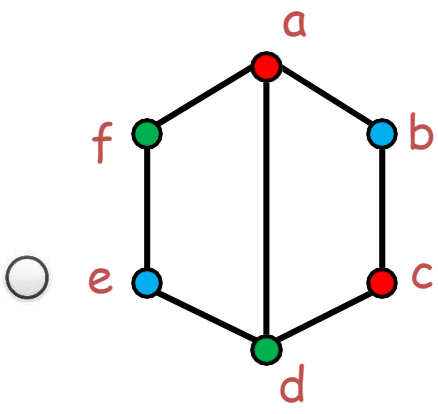
- ☐ A and B
- ☐ A and C
- ☐ B and C
- ☐ C and D

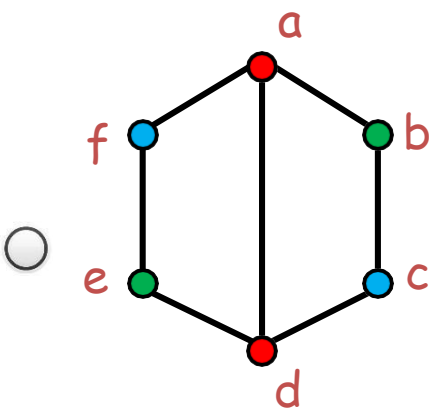


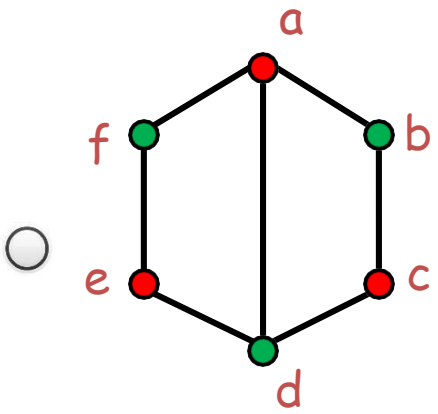
Question?

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



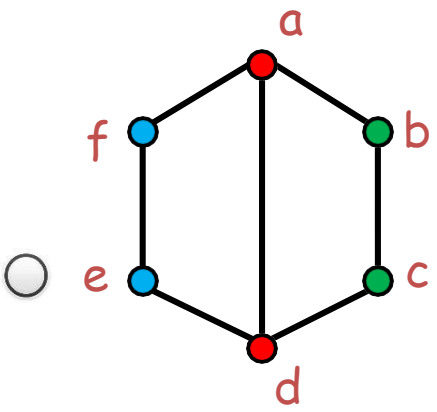


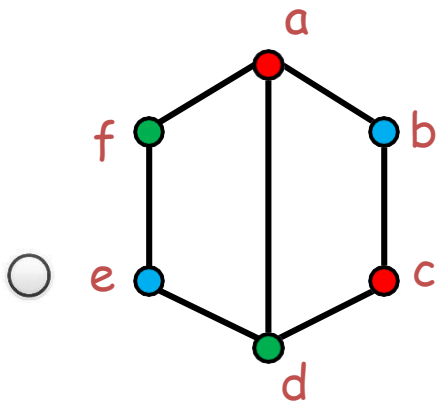


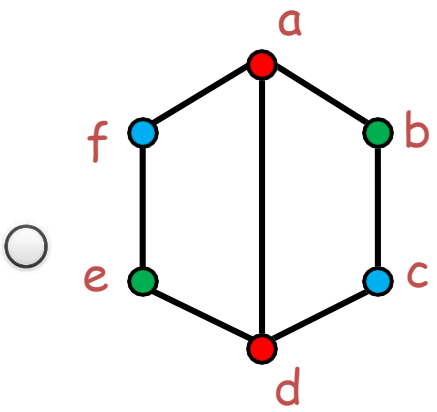


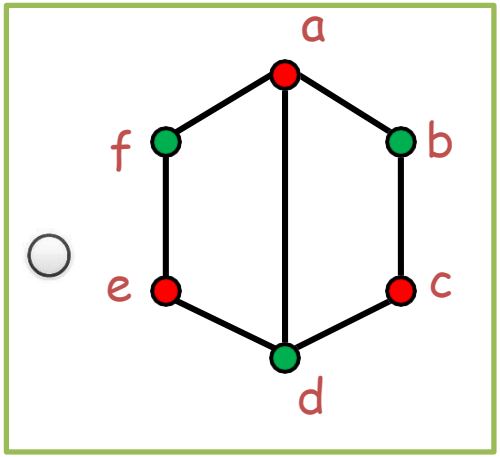
Answer!

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



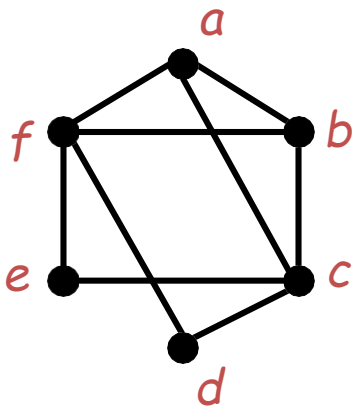






Question?

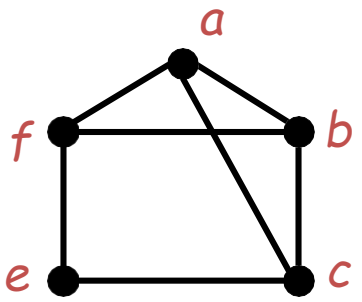
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Question?

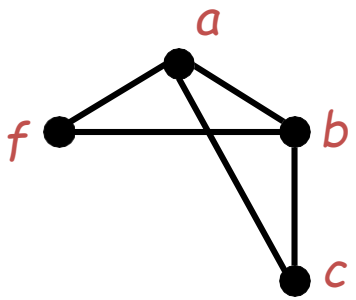
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Question?

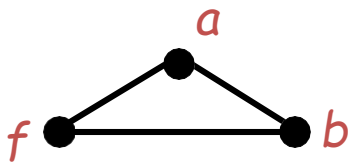
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Question?

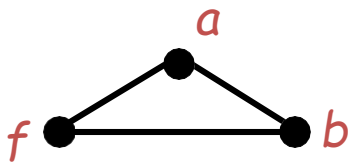
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Question?

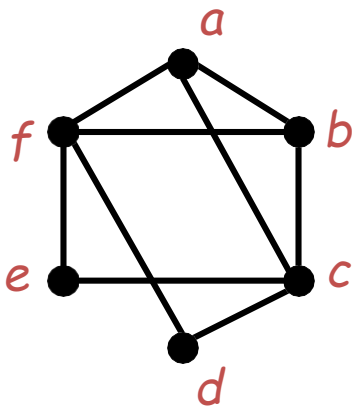
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Question?

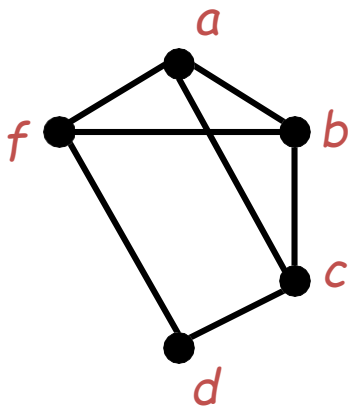
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Question?

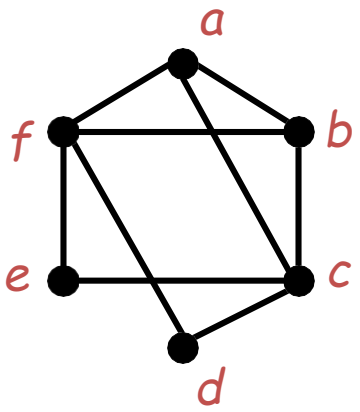
For the given RIG and k = 3, which of the following deletion orders are valid for the nodes of the given RIG?



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

Answer!

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



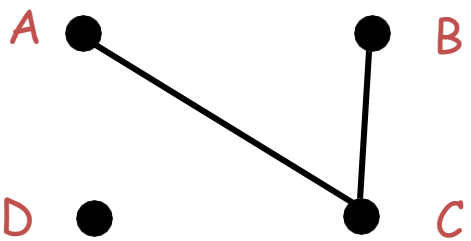
- {d, e, c, b, a, f}
- {e, f, a, b, c, d}
- {d, c, b, a, f , e}
- {d, e, b, c, a, f}

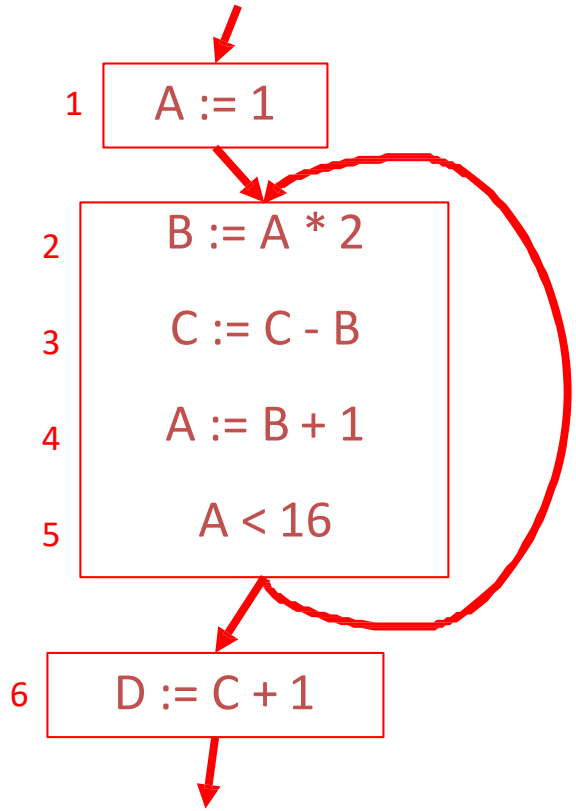
Question?

For the given code fragment and RIG, find the minimum cost spill. In this example, the cost of spilling a node is given by:

# of occurrences (use or definition)
- # of conflicts
+ 5 if the node corresponds to a variable used in a loop

- ☐ A
- ☐ B
- ☐ C
- ☐ D





Question?

For the given code fragment and RIG, find the minimum cost spill. In this example, the cost of spilling a node is given by:

# of occurrences (use or definition)
- # of conflicts
+ 5 if the node corresponds to a variable used in a loop

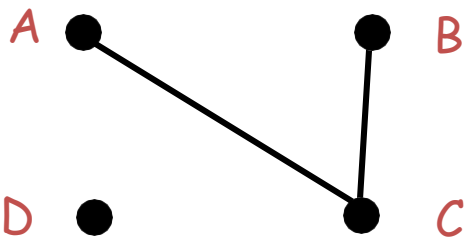
cost

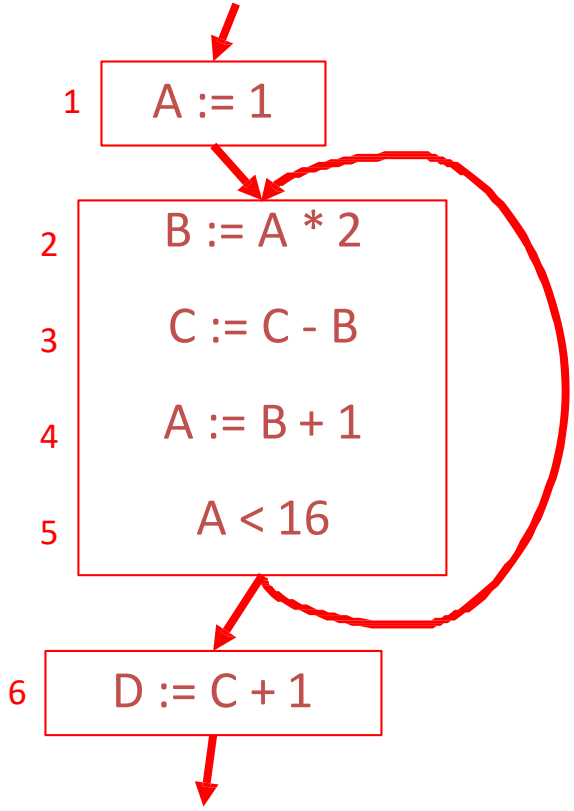
4-1+15=18 ☐ A

3-1+15=17 ☐ B

3-2+10=11 ☐ C

1-0+0=1 ☐ D

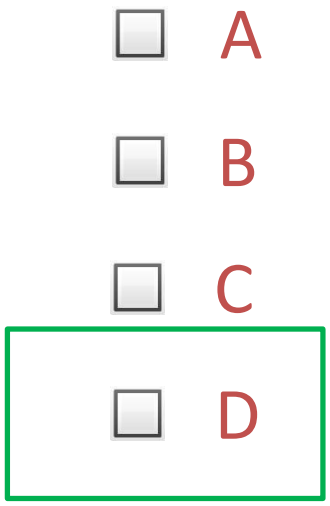


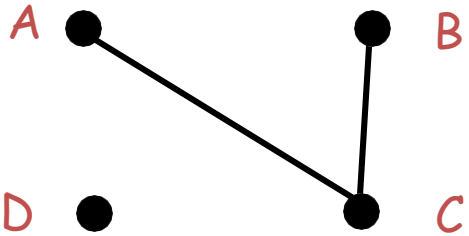


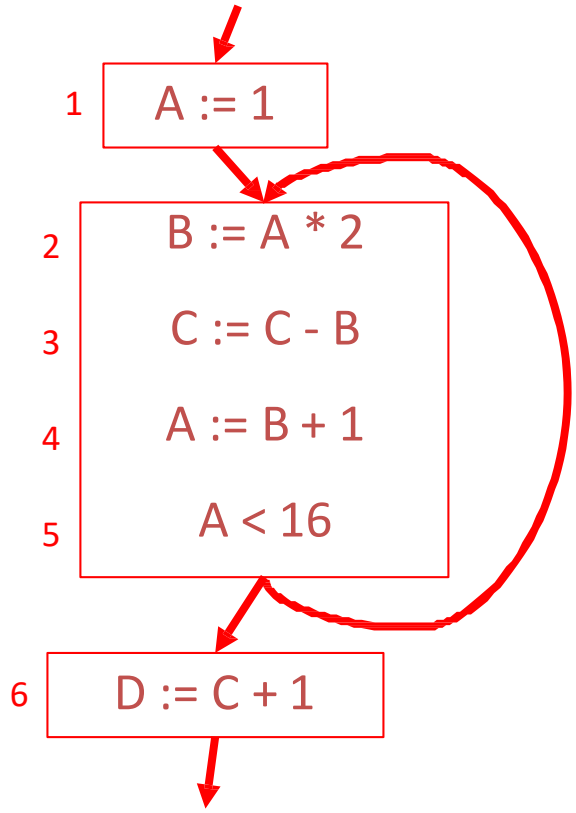
Answer!

For the given code fragment and RIG, find the minimum cost spill. In this example, the cost of spilling a node is given by:

# of occurrences (use or definition)
- # of conflicts
+ 5 if the node corresponds to a variable used in a loop







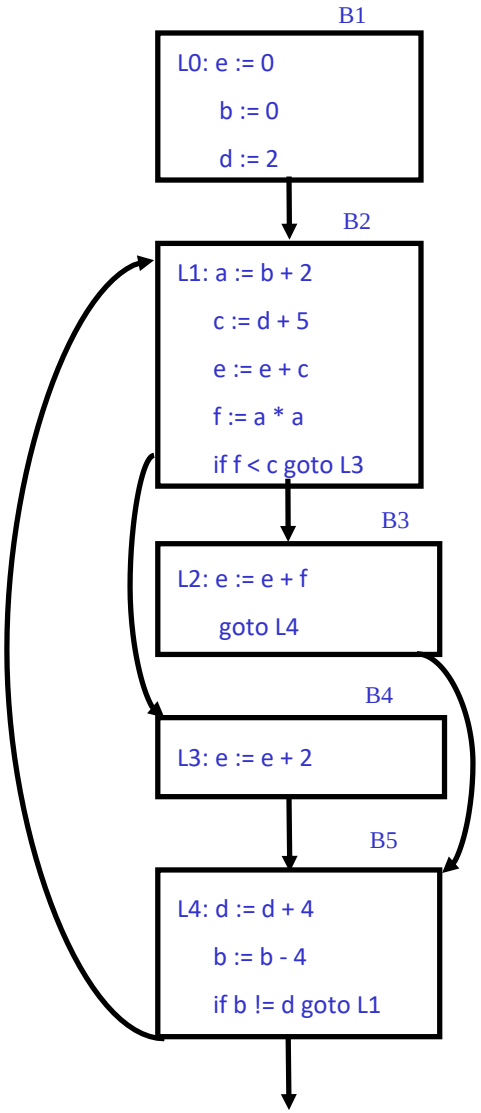
Question?

Assume at the end of following three-address code fragment, only **e** is live.

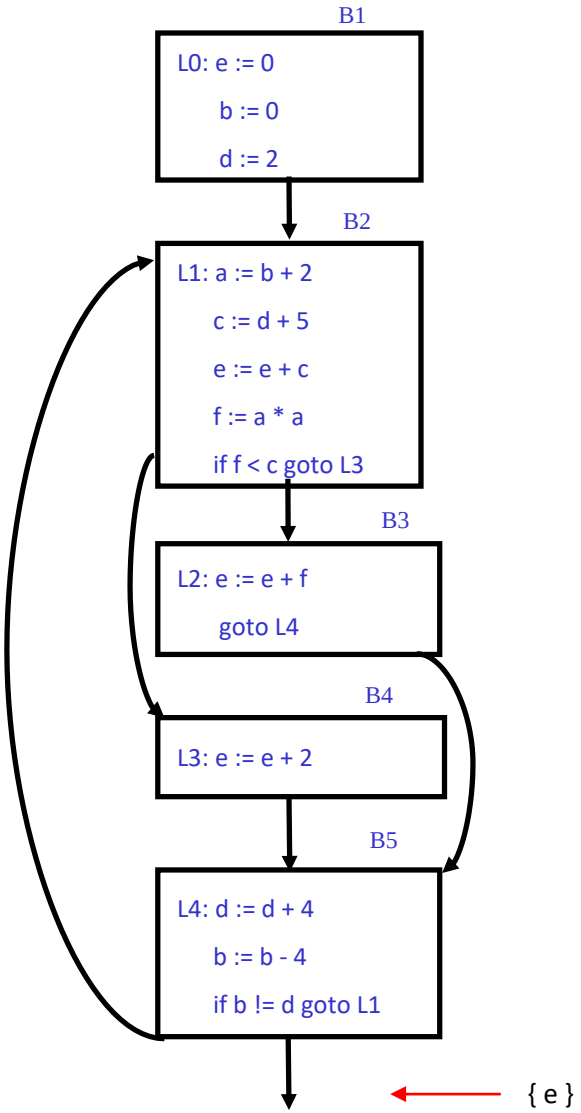
1. Draw the control flow graph of this code
2. Perform Liveness analysis
3. Draw the register inference graph (RIG), and
4. Determine at least how many registers are required for executing this code.
5. Then rewrite the code by spilling variable **b**.

```
L0: e := 0
    b := 0
    d := 2
L1: a := b + 2
    c := d + 5
    e := e + c
    f := a * a
    if f < c goto L3
L2: e := e + f
    goto L4
L3: e := e + 2
L4: d := d + 4
    b := b - 4
    if b != d goto L1
```

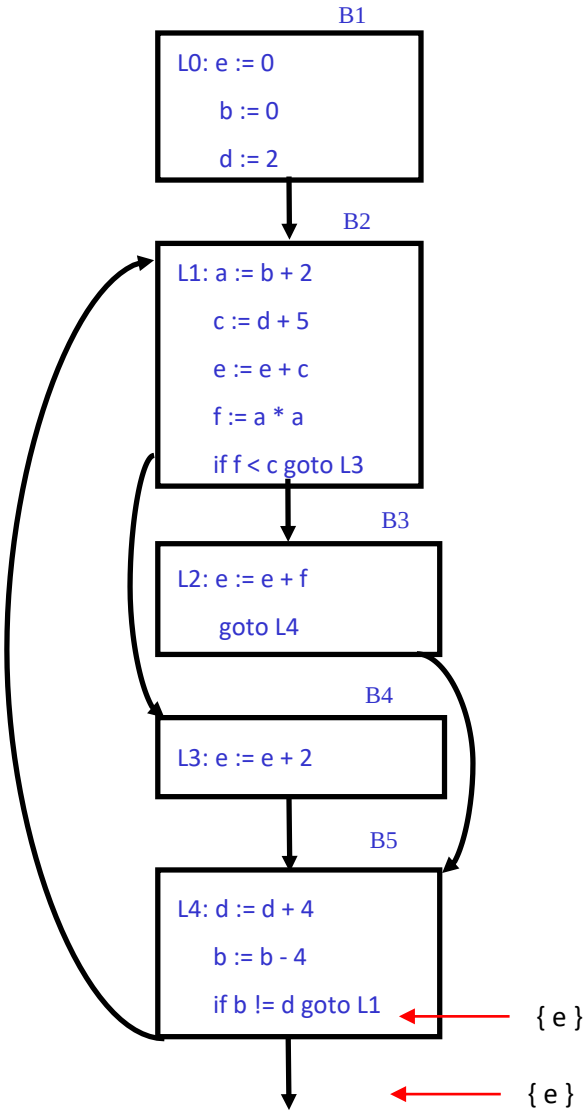
Control Flow Graph



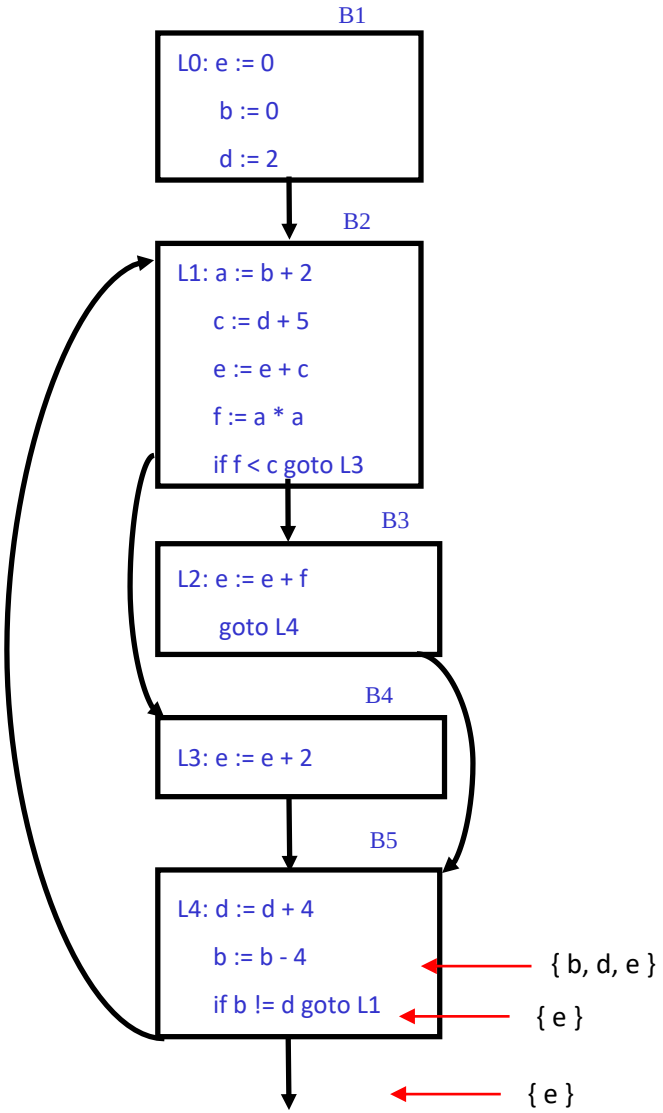
Liveness Analysis



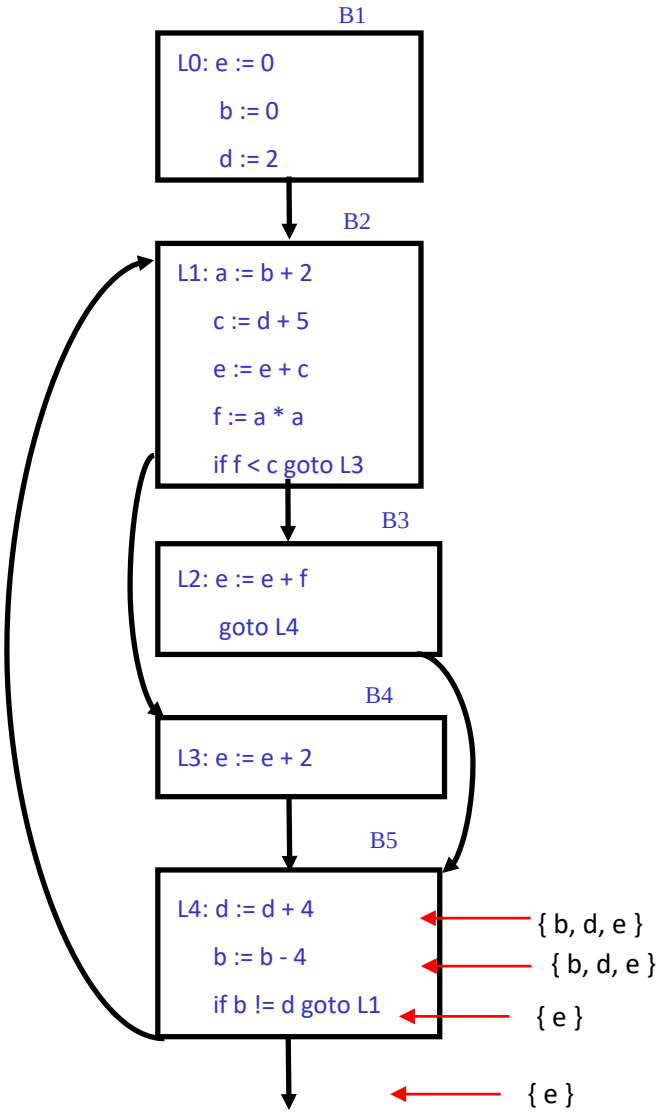
Liveness Analysis



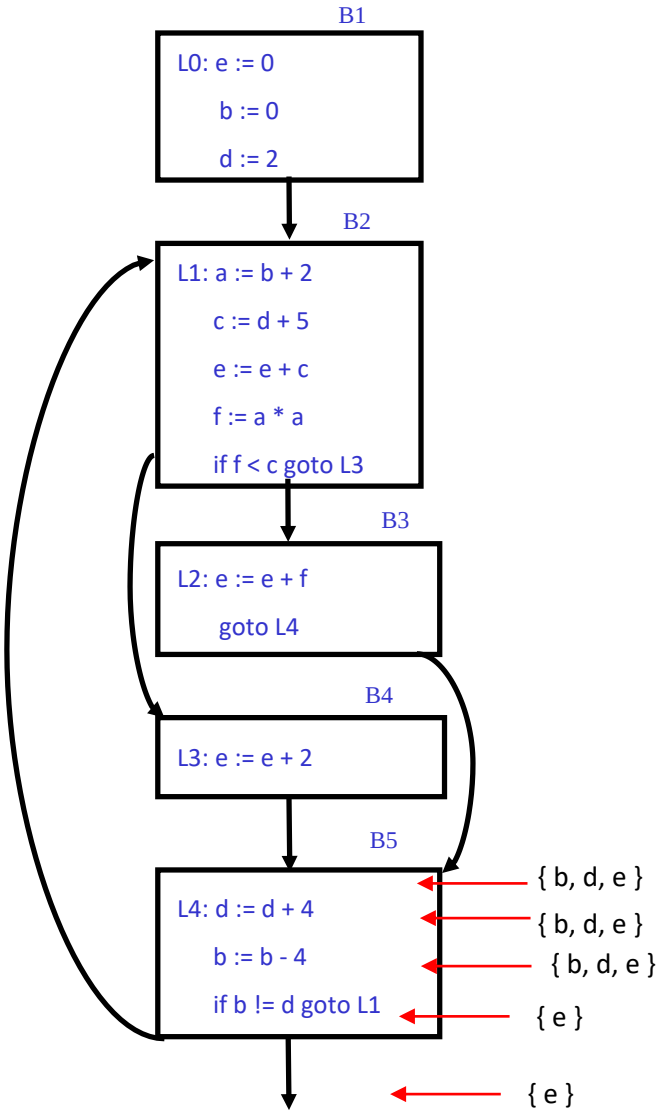
Liveness Analysis



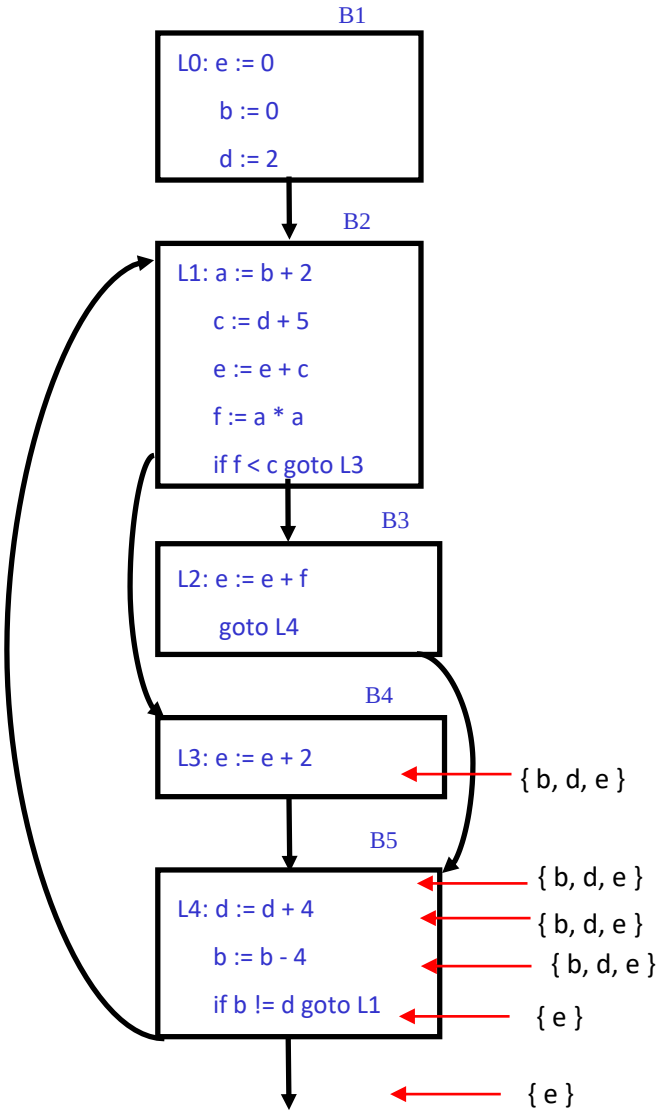
Liveness Analysis



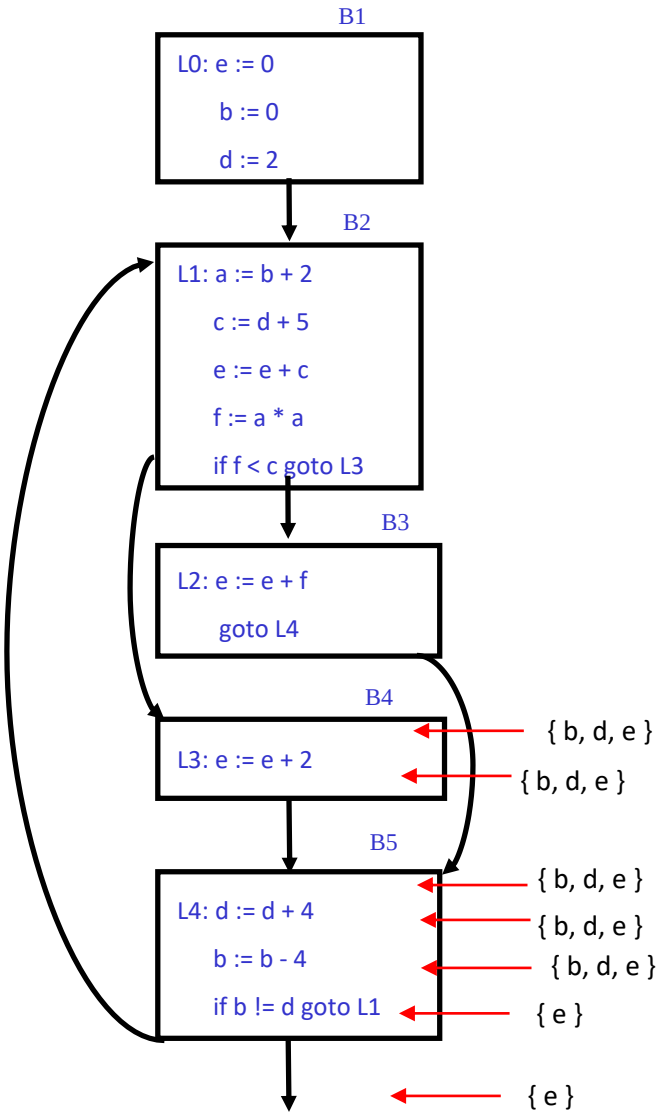
Liveness Analysis



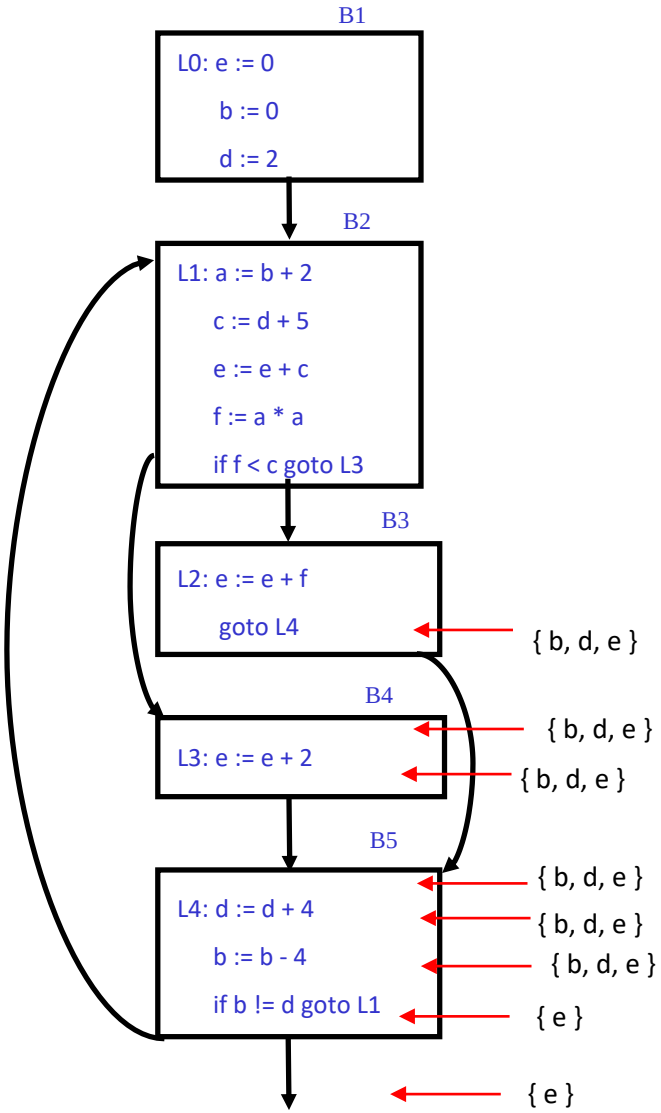
Liveness Analysis



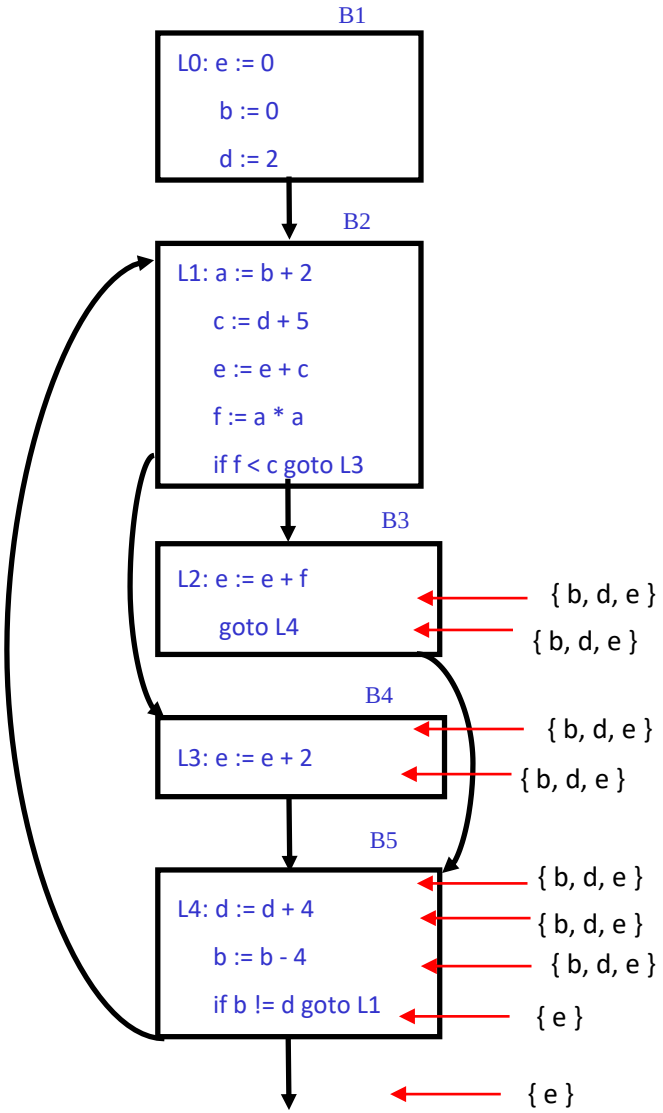
Liveness Analysis



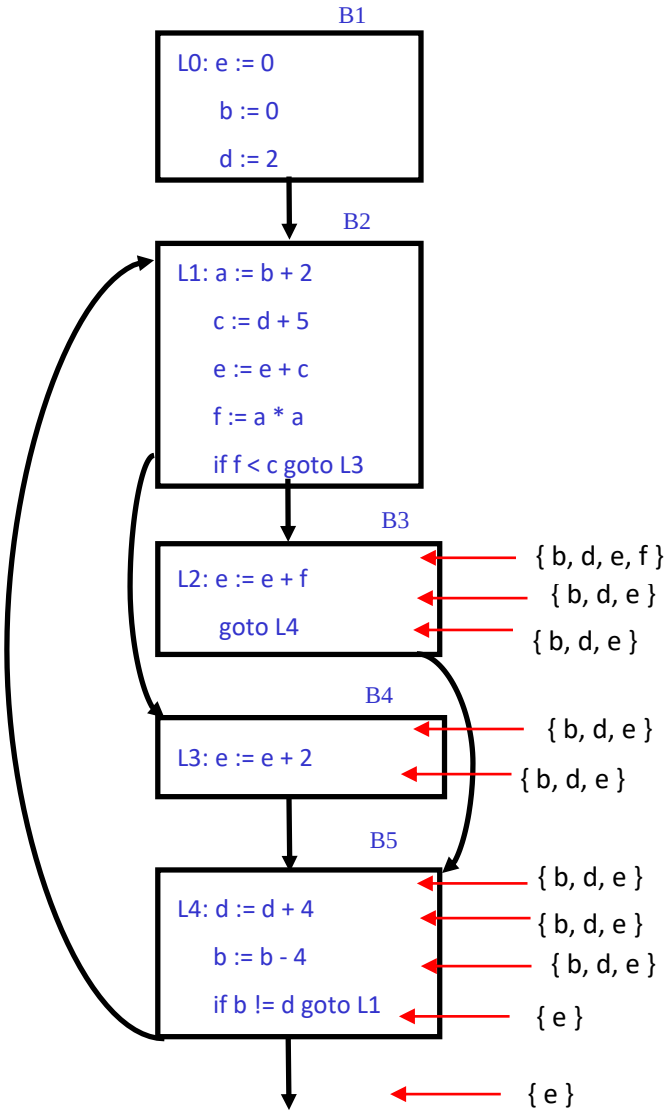
Liveness Analysis



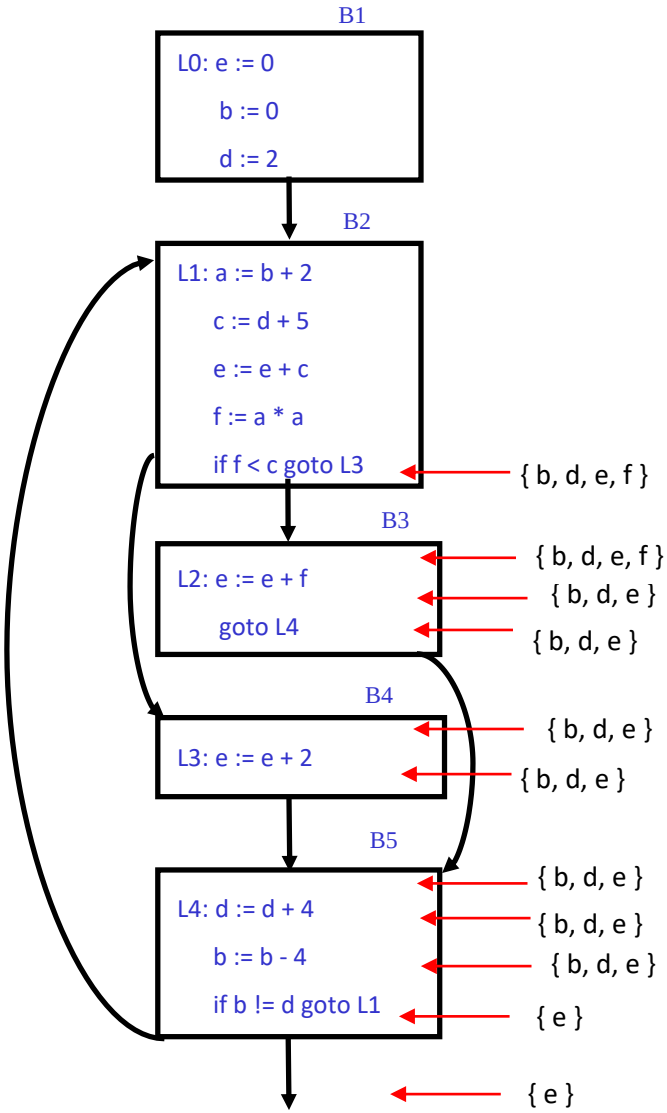
Liveness Analysis



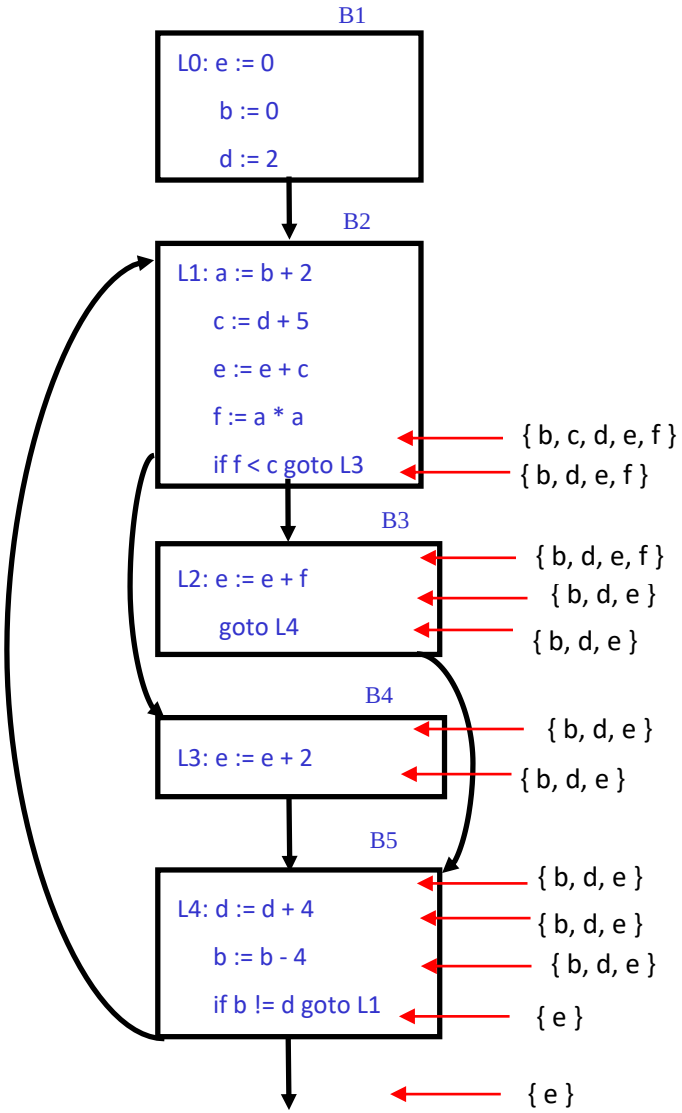
Liveness Analysis



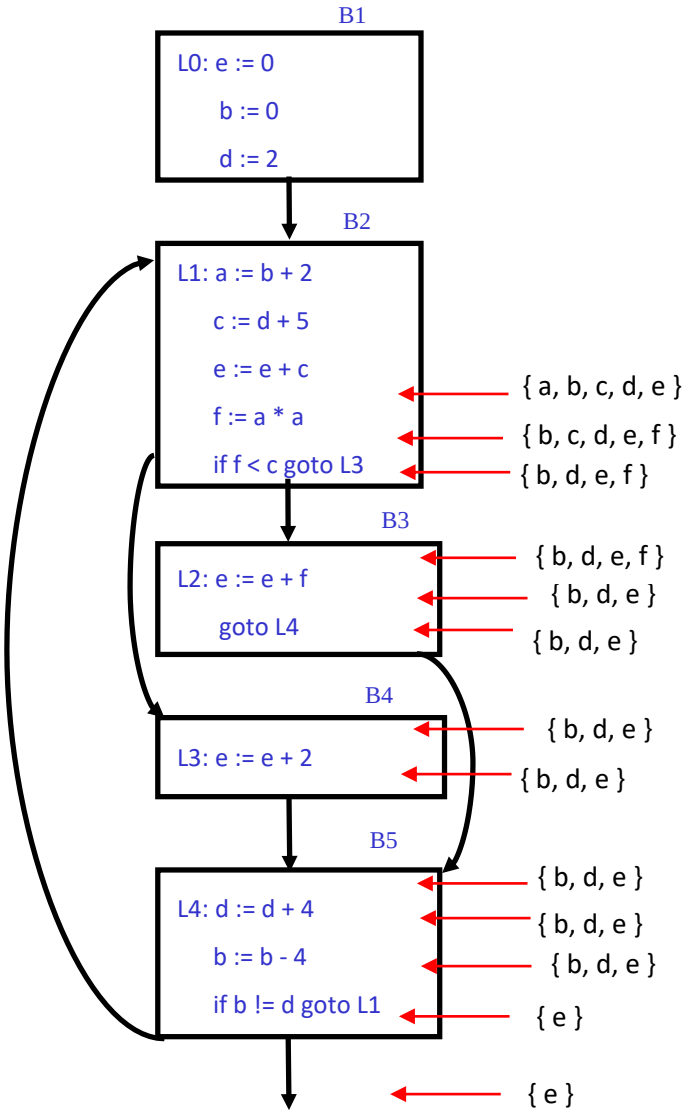
Liveness Analysis



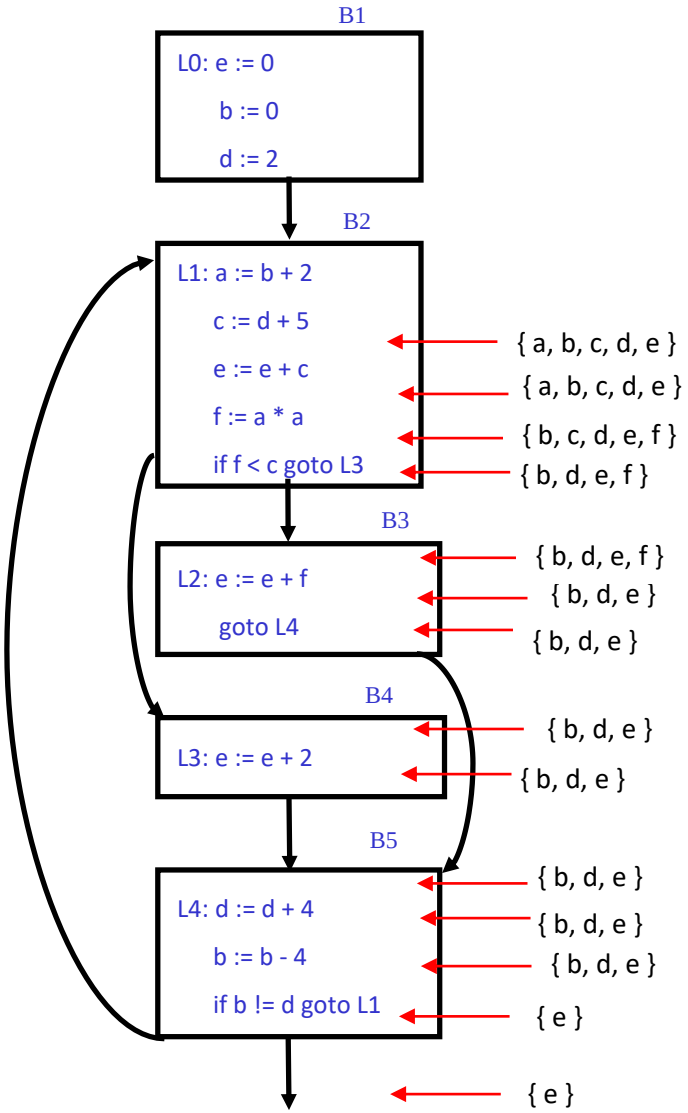
Liveness Analysis



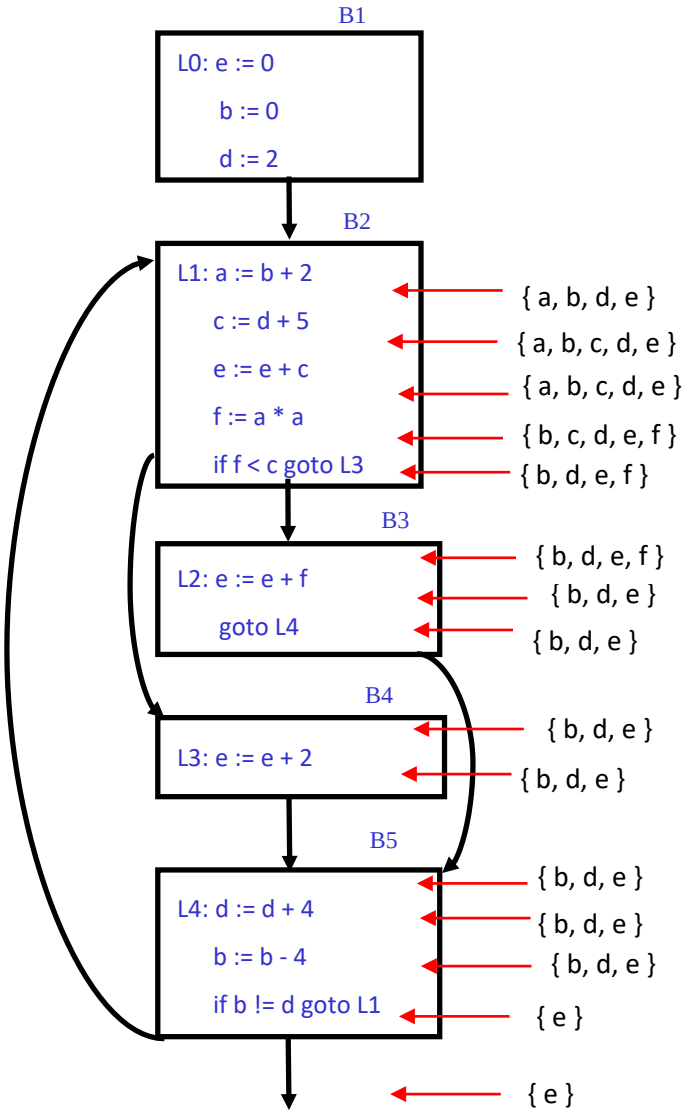
Liveness Analysis



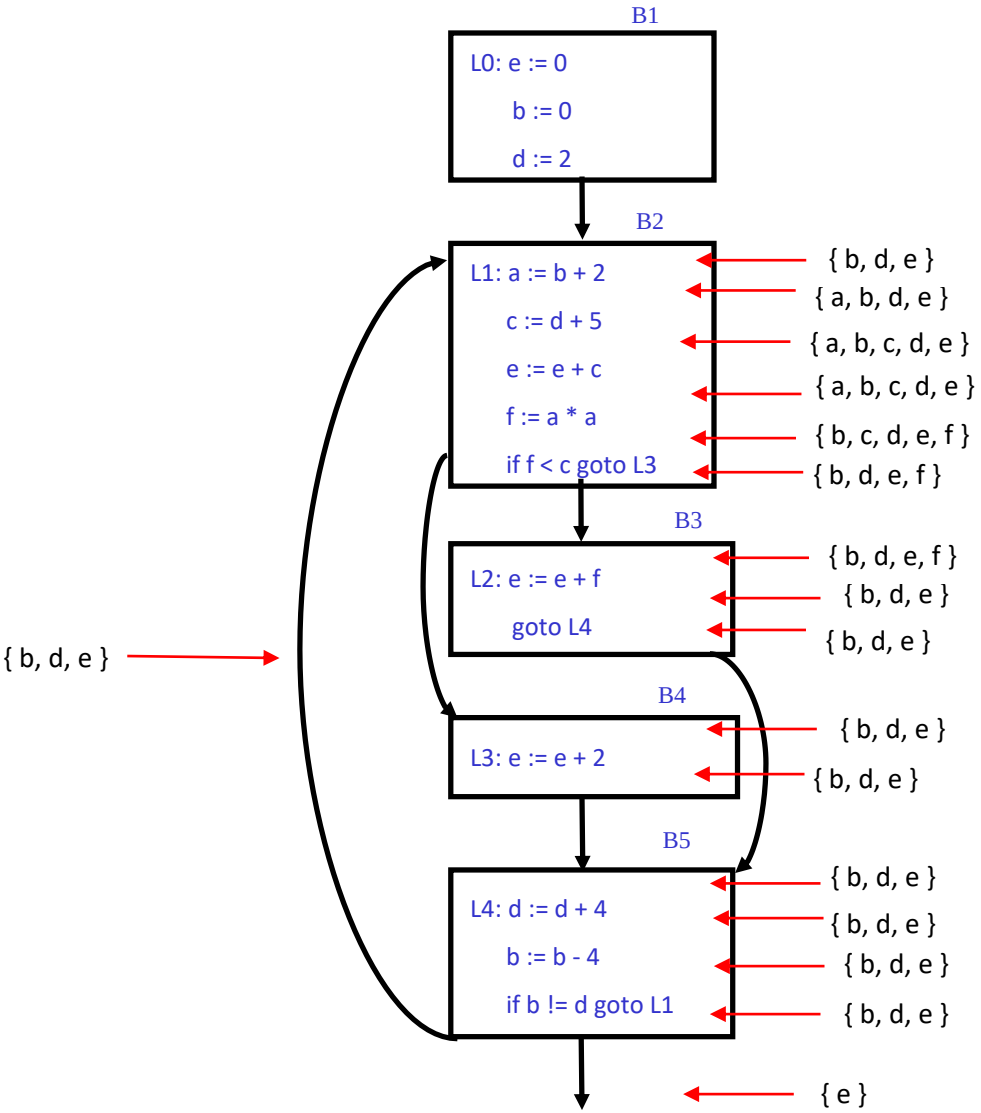
Liveness Analysis



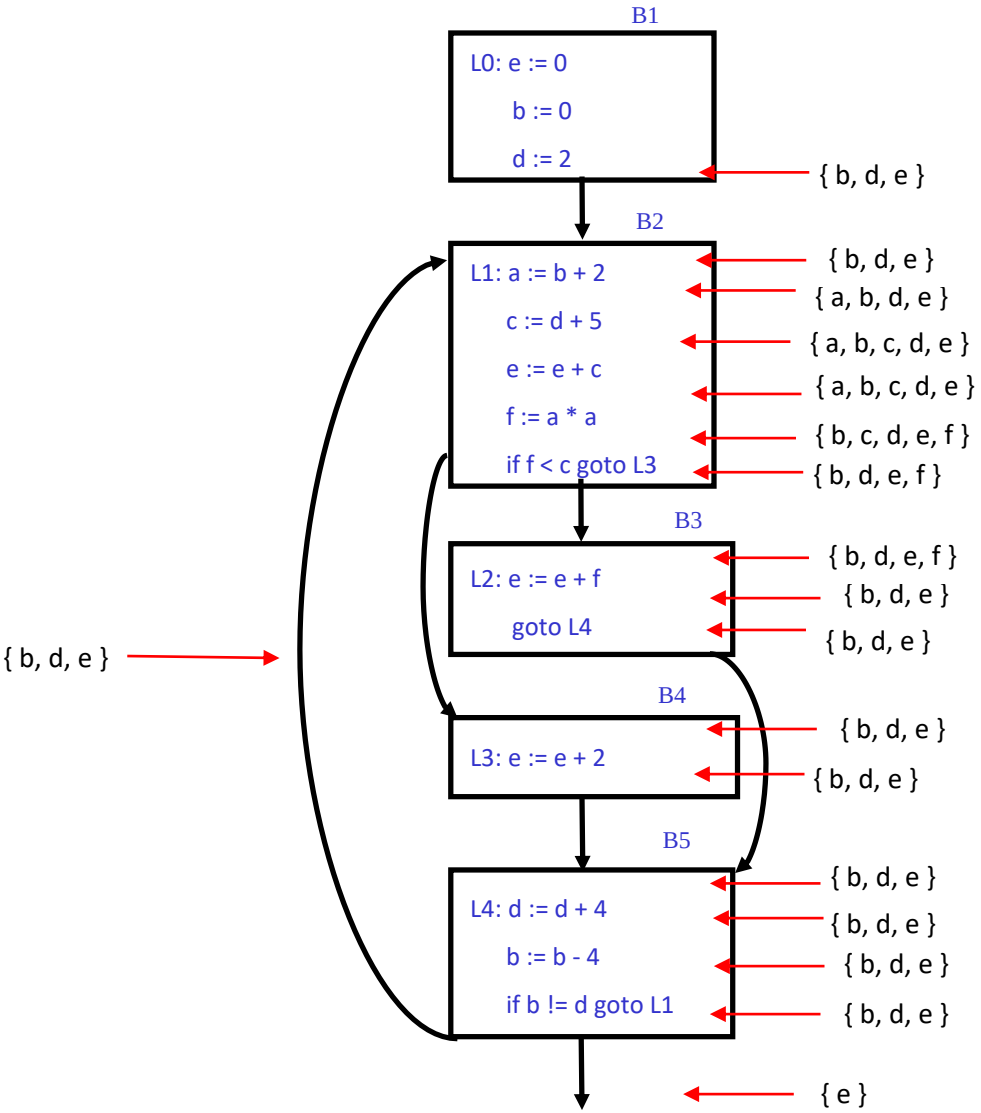
Liveness Analysis



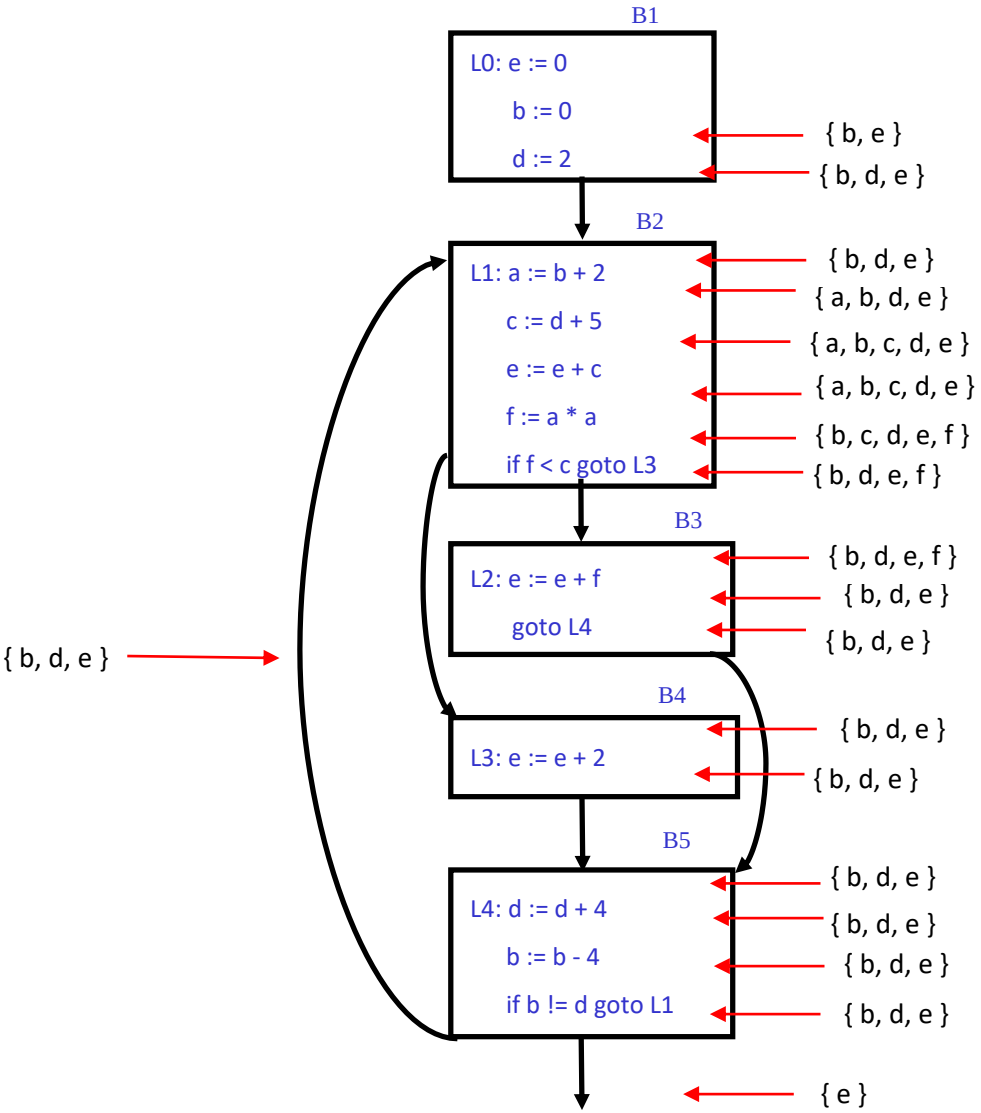
Liveness Analysis



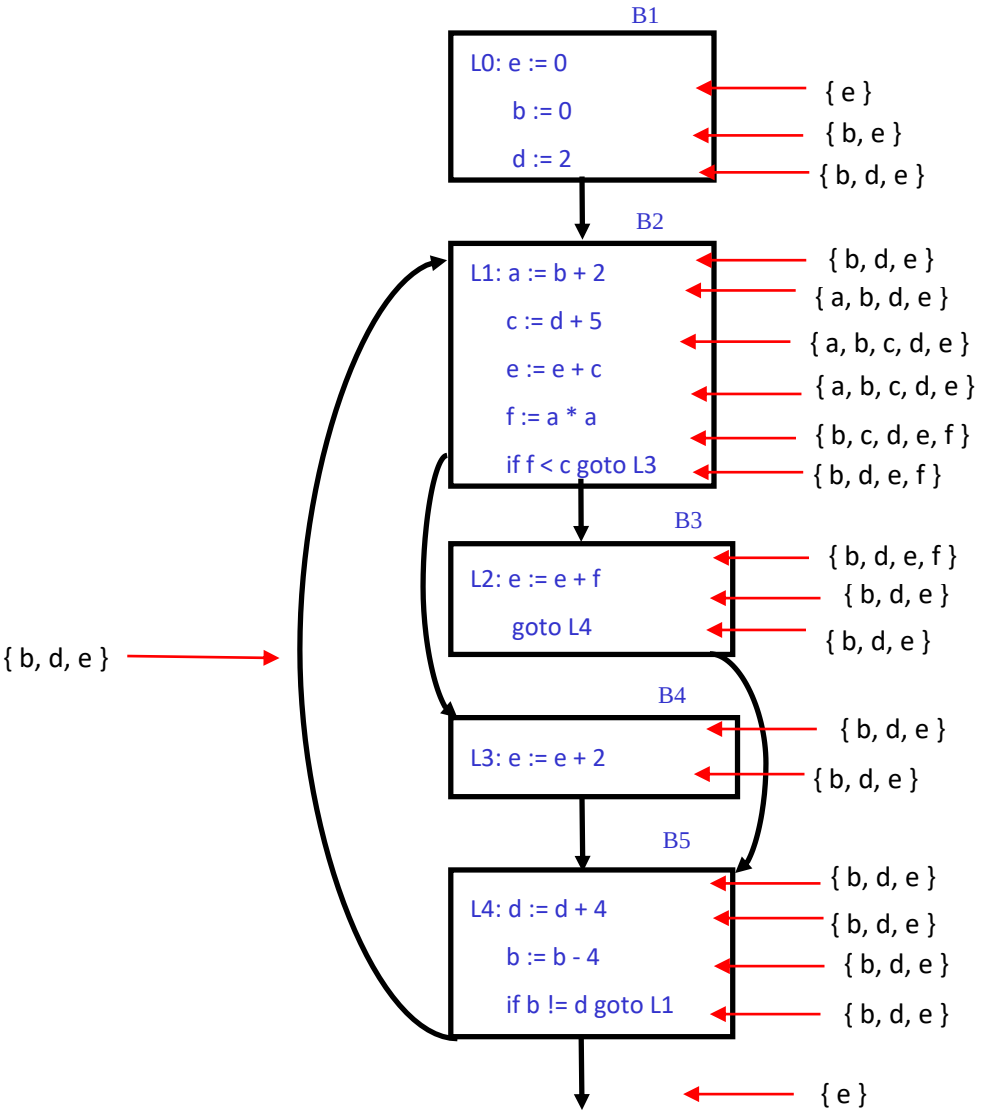
Liveness Analysis



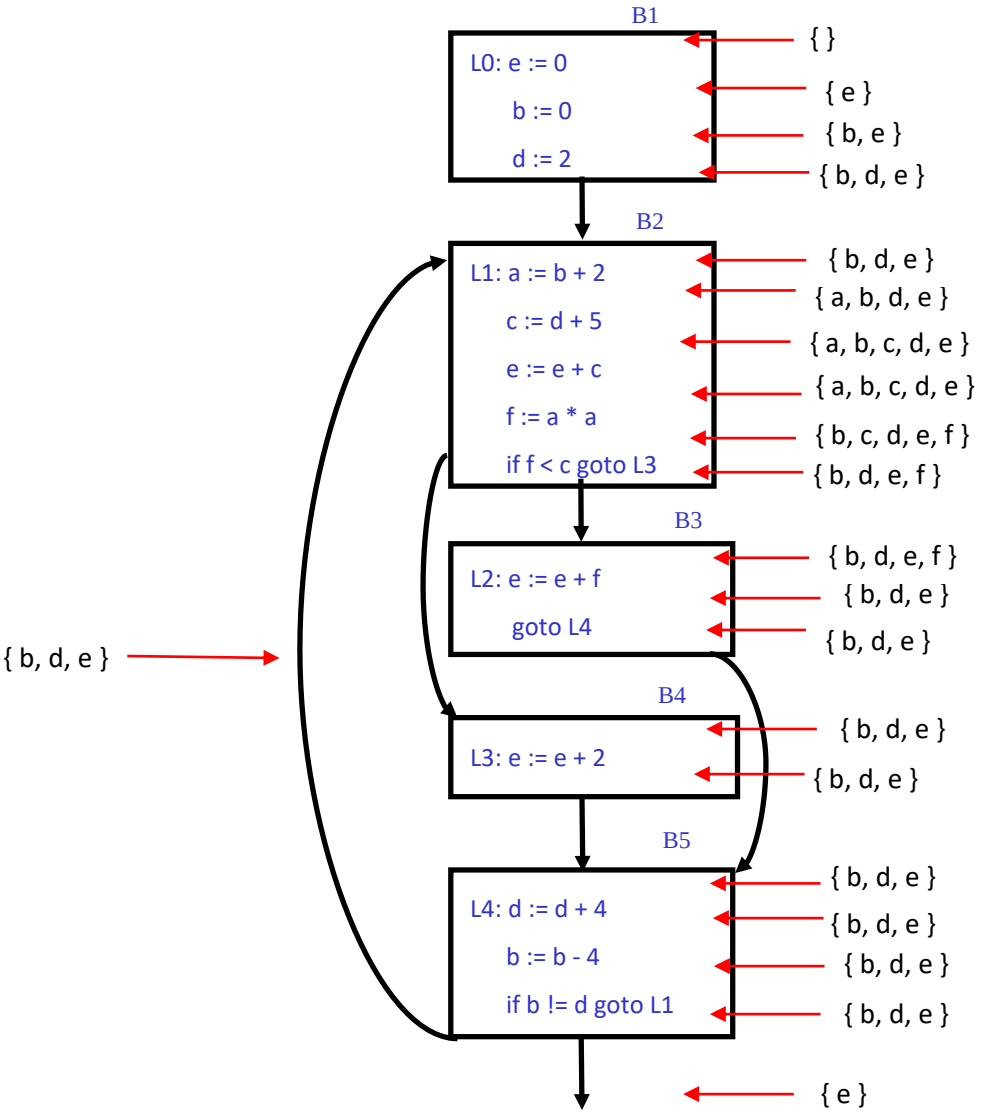
Liveness Analysis



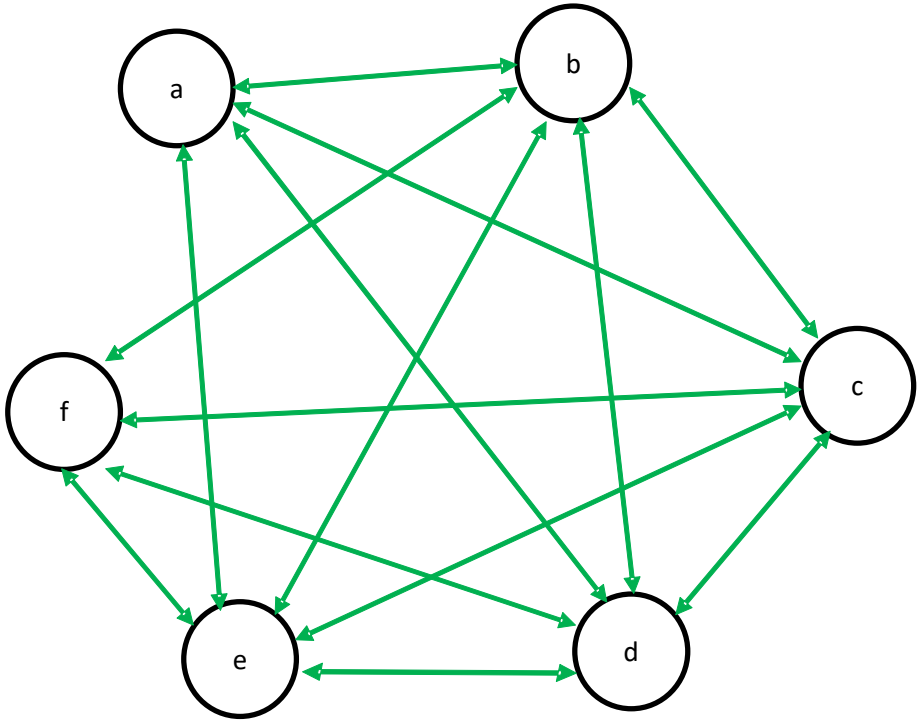
Liveness Analysis



Liveness Analysis



Register Interference Graph



Therefore, we need at least five registers to keep the value of these variables. In other words, only two variables (i.e., **a** and **f)** can safely be saved in the same register.

Code after spilling b

```
L0: e := 0
    b := 0
    d := 2
L1: a := b + 2
    c := d + 5
    e := e + c
    f := a * a
    if f < c goto L3
L2: e := e + f
    goto L4
L3: e := e + 2
L4: d := d + 4
    b := b - 4
    if b != d goto L1
```

```
L0: e := 0
    b1 := 0
    store b1, ba
    d := 2
L1: b2 := load ba
    a := b2 + 2
    c := d + 5
    e := e + c
    f := a * a
    if f < c goto L3
L2: e := e + f
    goto L4
L3: e := e + 2
L4: d := d + 4
    b3:= load ba
    b3 := b3 - 4
    store b3, ba
    if b3 != d goto L1
```

Liveness Analysis

Now, we need at least four registers to keep the value of these variables.