Global Register Allocation with Graph Coloring

- When a register is needed but all available registers are in use, the content of one of the used registers must be stored (spilled) to free a register.
- Graph coloring allocates registers and attempts to minimize the cost of spills.
- Build a conflict graph (interference graph).
- Find a $k$-coloring for the graph, with $k$ the number of registers.
Register Allocation with Graph Coloring: Example

```plaintext
a := read();
b := read();
c := read();
a := a + b + c;
if (a < 10) {
    d := c + 8;
    write(c);
} else if (a < 20) {
    e := 10;
    d := e + a;
    write(e);
} else {
    f := 12;
    d := f + a;
    write(f);
}
write(d);
```
Register Allocation with Graph Coloring: Live Ranges

Interference graph: connected vars have overlapping ranges

Live range of b
Register Allocation with Graph Coloring: Solution

Interference graph

Solve

Three registers:
\[
\begin{align*}
a &= r2 \\
b &= r3 \\
c &= r1 \\
d &= r2 \\
e &= r1 \\
f &= r1
\end{align*}
\]

```plaintext
r2 := read();
r3 := read();
r1 := read();
r2 := r2 + r3 + r1;
if (r2 < 10) {
    r2 := r1 + 8;
    write(r1);
} else if (r2 < 20) {
    r1 := 10;
    r2 := r1 + r2;
    write(r1);
} else {
    r1 := 12;
    r2 := r1 + r2;
    write(r1);
}
write(r2);
```