Dependence Analysis
Motivating question

• Can the loops on the right be run in parallel?
  • *i.e.*, can different processors run different iterations in parallel?

• What needs to be true for a loop to be parallelizable?
  • Iterations cannot interfere with each other
  • No *dependence* between iterations

```c
for (i = 1; i < N; i++) {
    a[i] = b[i];
    c[i] = a[i - 1];
}
```

```c
for (i = 1; i < N; i++) {
    a[i] = b[i];
    c[i] = a[i] + b[i - 1];
}
```
Dependences

for (i = 1; i < N; i++) {
    a[i] = b[i];
    c[i] = a[i - 1];
}

- A **flow dependence** occurs when one iteration writes a location that a later iteration reads

\[
\begin{array}{cccccc}
  i = 1 & i = 2 & i = 3 & i = 4 & i = 5 \\
  R(b[1]) & R(b[2]) & R(b[3]) & R(b[4]) & R(b[5]) \\
  W(a[1]) & W(a[2]) & W(a[3]) & W(a[4]) & W(a[5]) \\
  R(a[0]) & R(a[1]) & R(a[2]) & R(a[3]) & R(a[4]) \\
  W(c[1]) & W(c[2]) & W(c[3]) & W(c[4]) & W(c[5]) \\
\end{array}
\]
Running a loop in parallel

- If there is a dependence in a loop, we cannot guarantee that the loop will run correctly in parallel
  - What if the iterations run out of order?
    - Might read from a location before the correct value was written to it
  - What if the iterations do not run in lock-step?
    - Same problem!
Other kinds of dependence

- **Anti dependence** – When an iteration *reads* a location that a later iteration *writes* (why is this a problem?)

```c
for (i = 1; i < N; i++) {
    a[i - 1] = b[i];
    c[i] = a[i];
}
```

- **Output dependence** – When an iteration *writes* a location that a later iteration *writes* (why is this a problem?)

```c
for (i = 1; i < N; i++) {
    a[i] = b[i];
    a[i + 1] = c[i];
}
```
Data dependence concepts

- Dependence *source* is the earlier statement (the statement at the tail of the dependence arrow)

- Dependence *sink* is the later statement (the statement at the head of the dependence arrow)

Dependences can only go forward in time: always from an earlier iteration to a later iteration.
Using dependences

• If there are no dependences, we can parallelize a loop
  • None of the iterations interfere with each other
• Can also use dependence information to drive other optimizations
  • Loop interchange
  • Loop fusion
• How do we represent dependences in loops?