Loop Invariants

☐ What are they?

Property — a relationship among the variables of loop
TRUE — (1) after initialization
(2) after every iteration

☐ Why do we care about them?

AVOID BUGS

☐ How do we use them?

(1) After the fact — write code & then find an invariant
(2) Drive the implementation — write an invariant &
then use it to set up the loop

☐ Examples:
Example problem:

\[ \text{int } n > 0 \quad \text{int } k > 1 \quad \text{factorsIn} \quad \# \text{ of times } k \text{ is a factor of } n \]

factors \( \text{In}(56, 2) \) is 3 because \( 2^4 = 2 \times 2 \times 2 \times 7 \)

3 factors of 2

Observation:

factors \( \text{In}(n, k) = \text{factorsIn} \left( \frac{n}{k}, k \right) + 1 \) if \( n \% k == 0 \)

\[
\begin{align*}
\text{int factorsIn(int n, int k) \{} \\
\text{if (n \% k != 0)} \\
\text{return 0;} \\
\text{else} \\
\text{return 1 + factorsIn(n/k, k);} \\
\};
\end{align*}
\]

\[
\begin{align*}
\text{int factorsIn(int n, int k) \{} \\
\text{int count = 0;} \\
\text{while (n \% k == 0) \{} \\
\text{count++} \\
\text{n = n/k;} \\
\}; \\
\text{return count;} \\
\};
\end{align*}
\]
```c
int factors_in(int n, int k) {
    int count = 0;
    while (n % k == 0) {
        count++;
        n = n / k;
    }
    return count;
}
```

Let $N$ be the original value of $N$

<table>
<thead>
<tr>
<th>$N$</th>
<th>$n$</th>
<th>$k$</th>
<th>$\text{count}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>56</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>56</td>
<td>28</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>56</td>
<td>14</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>56</td>
<td>7</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

**Invariants**

1. Initially: \( \text{count} = 0 \)
2. Each iteration preserves it
3. On termination, \( n \% k != 0 \)

Assuming \( n \% k == 0 \):

- \( count++ \)
- \( n / k == k \)
- \( count \times k == N \)
- \( \Rightarrow \text{count has correct final value} \)
Example invariants:

1. \( x == 0 \)  \quad \text{On termination, } \ l == n

2. \( \text{lowest} == \text{minimum among } a[0], a[1], \ldots, a[\text{current}-1] \)  \quad \text{On termination, } \ \text{current} == a.\text{length}

3. \( \text{sum} == \text{sum of } a[0], a[1], \ldots, a[\text{c}-1] \)  \quad \text{On termination, } \ c == a.\text{length}

4. \( n^x == p \)  \quad \text{On termination, } \ x == k