

# Computer Programming

Dr. Deepak B Phatak  
Dr. Supratik Chakraborty  
Department of Computer Science and Engineering  
IIT Bombay

Session: Reasoning about loops

# Quick Recap of Relevant Topics

---



- Sequential and conditional execution of statements
- Iteration/looping constructs
- Solving simple problems with iteration constructs in C++

# Overview of This Lecture

---



- Reasoning about loops as a design and post-design activity

Pre-conditions

Post-conditions

Loop invariants

Loop variants

# Reasoning About Loops



- Loop: Part of program with an iterative construct
  - Won't distinguish between “for ...”, “while ...”, “do ... while ...” loops for this discussion
- What does a loop compute ?
  - Enter the loop with some relation among variables : **pre-condition**
  - Exit the loop with some relation among variables: **post-condition**
  - Loop incrementally changes variables such that we move pre-condition to post-condition

# Recall: Min/Max Example



**Given positive integers  $m$  and  $n$ , find  $3^{\min(m, n)}$  and  $2^{\max(m, n)}$**

```
int minMN = 0, maxMN = 0, threeRaisedMin = 1, twoRaisedMax = 1;
int m, n, i, j;
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--) // Iterate max(m, n) times
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3; // Conditionally iterate min(m,n) times
}
maxMN++; twoRaisedMax *= 2; // Executed max(m, n) times
}
```

# Recall: Min/Max Example



**Given positive integers  $m$  and  $n$ , find  $3^{\min(m, n)}$  and  $2^{\max(m, n)}$**

```
// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$   
//                 $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$ 
```

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--) // Iterate  $\max(m, n)$  times  
{ if ((i >= 1) && (j >= 1)) {  
    minMN++; threeRaisedMin *= 3; // Conditionally iterate  $\min(m, n)$  times  
}  
    maxMN++; twoRaisedMax *= 2; // Executed  $\max(m, n)$  times  
}
```

```
// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$   
//                 $\text{twoRaisedMax} = 2^{\max(m, n)}, \text{threeRaisedMin} = 3^{\min(m, n)}$ 
```

# Reasoning About Loops



**Given positive integers  $m$  and  $n$ , find  $3^{\min(m, n)}$  and  $2^{\max(m, n)}$**

**// PRECONDITION: integers  $m \geq 1$ ,  $n \geq 1$ ,  $\text{minMN} = 0$ ,  $\text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1$ ,  $\text{threeRaisedMin} = 1$**

Loop incrementally changes variables such that starting from pre-condition, eventually post-condition holds

**// POSTCONDITION:  $\text{minMN} = \min(m, n)$ ,  $\text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m, n)}$ ,  $\text{threeRaisedMin} = 3^{\min(m, n)}$**

# Designing Loops: A Design Activity



**Given positive integers  $m$  and  $n$ , find  $3^{\min(m, n)}$  and  $2^{\max(m, n)}$**

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//**

**twoRaisedMax = 1, threeRaisedMin = 1**

How do we incrementally  
change variables to effect  
transformation from  
pre-condition to post-condition?

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//**

**twoRaisedMax =  $2^{\max(m, n)}$ , threeRaisedMin =  $3^{\min(m, n)}$**



# Verifying Loops: A Post-Design Activity



Given positive integers  $m$  and  $n$ , find  $3^{\min(m, n)}$  and  $2^{\max(m, n)}$

// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$

//

for ( $i = m, j = n; (i \geq 1 \ \&\& \ j \geq 1);$

{ if ( $(i \geq 1) \ \&\& \ (j \geq 1)$

minMN++;

}

maxMN++;

}

// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$

//

twoRaisedMax =  $2^{\max(m, n)}$ , threeRaisedMin =  $3^{\min(m, n)}$

Does this loop really effect transformation from pre-condition to post-condition?

# Reasoning About Loops



- Designing and verifying loops requires similar kind of reasoning
- A loop iteratively transforms relations between variables such that
  - **Pre-condition** holds when we start iterating for first time
  - **Post-condition** holds when we exit loop
  - Pre-condition, post-condition special cases of relation that holds invariantly every time we are about to iterate: **loop-invariant**
  - **Desirable**: Integer valued “metric” (e.g. value of counter) monotonically changes towards fixed value : **loop-variant** (ensures loop termination)

# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:**

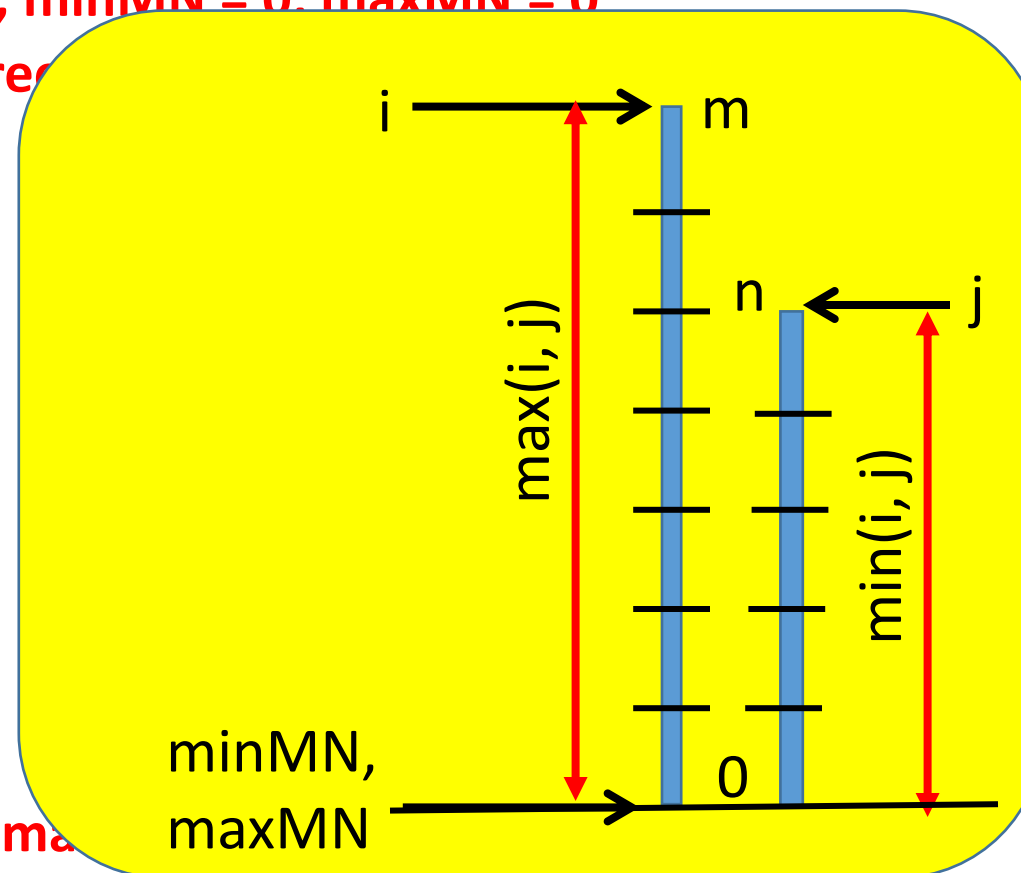
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m,n)}, \text{threeRaisedMin} = 3^{\min(m,n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:**

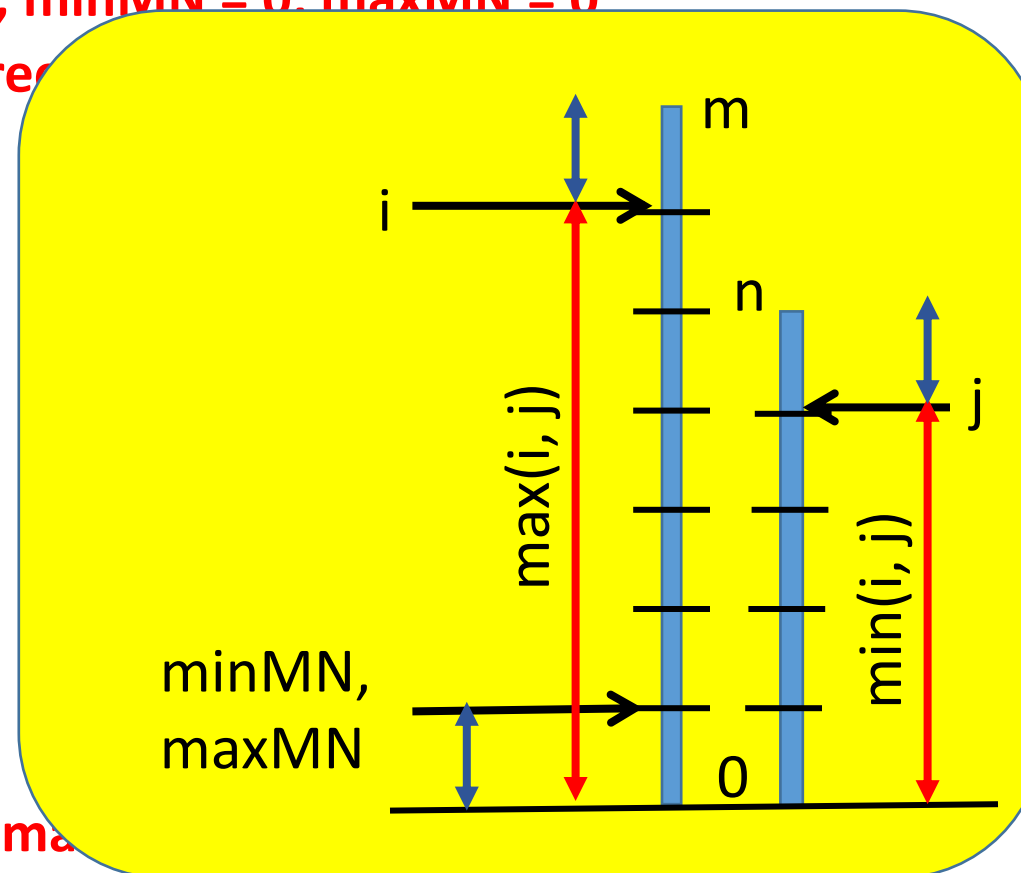
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m,n)}, \text{threeRaisedMin} = 3^{\min(m,n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:**

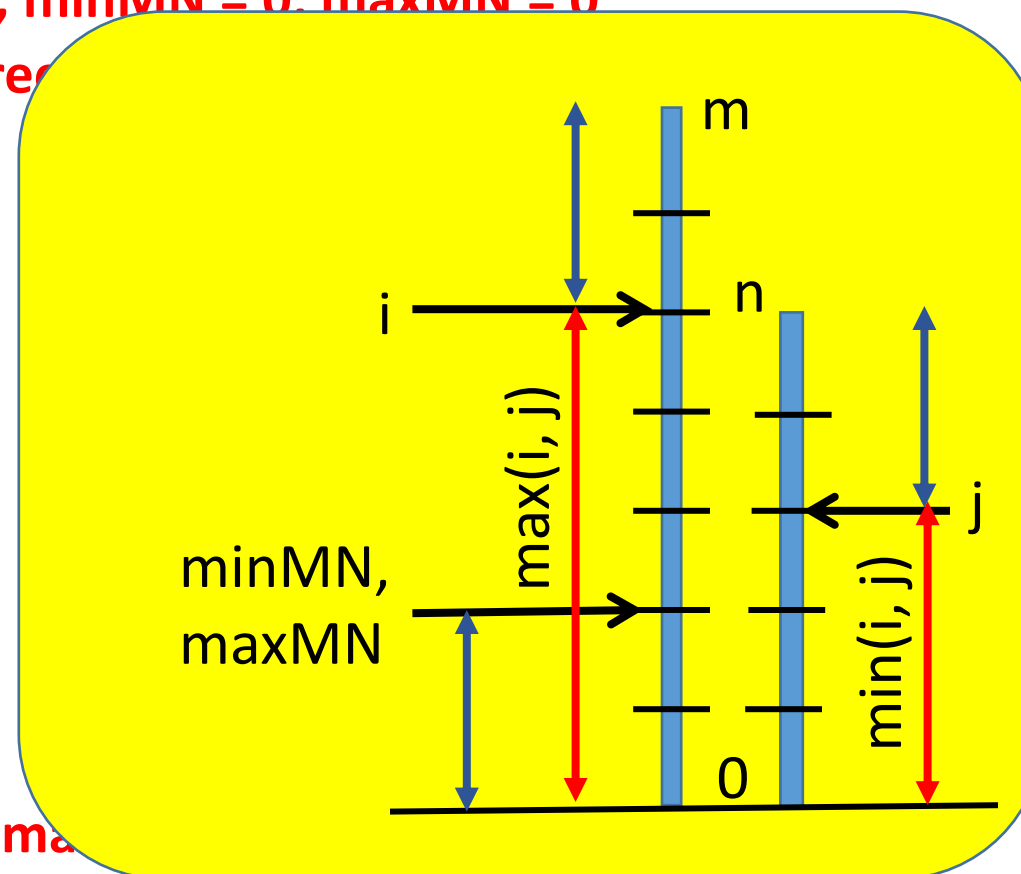
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m,n)}, \text{threeRaisedMin} = 3^{\min(m,n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:**

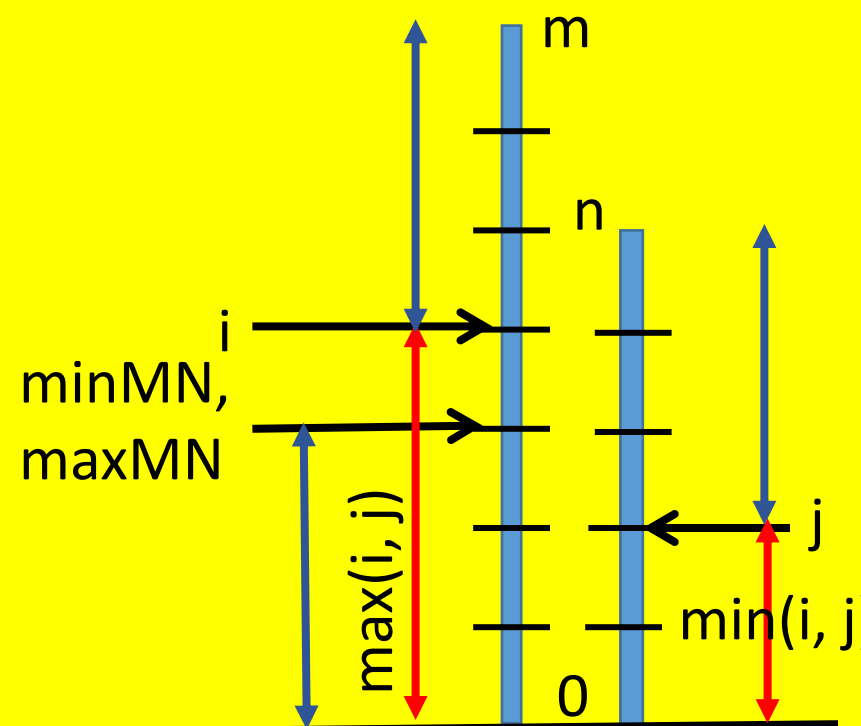
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m,n)}, \text{threeRaisedMin} = 3^{\min(m,n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:**

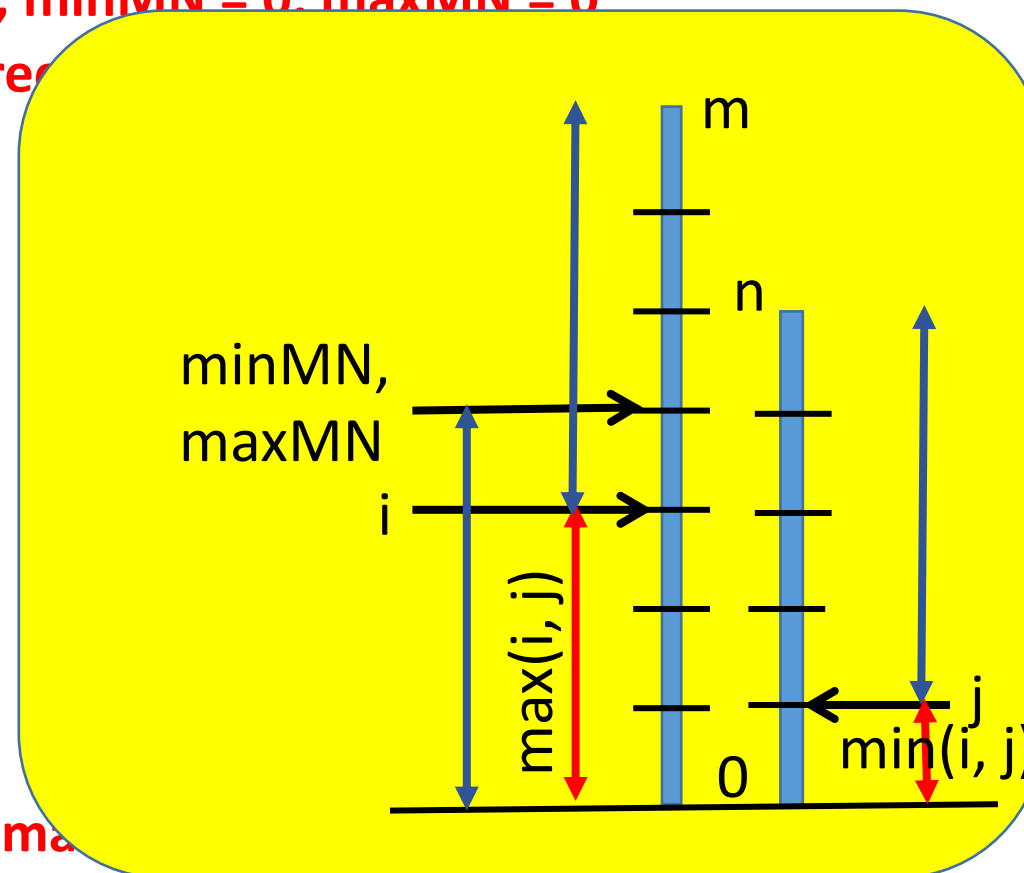
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m,n)}, \text{threeRaisedMin} = 3^{\min(m,n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:**

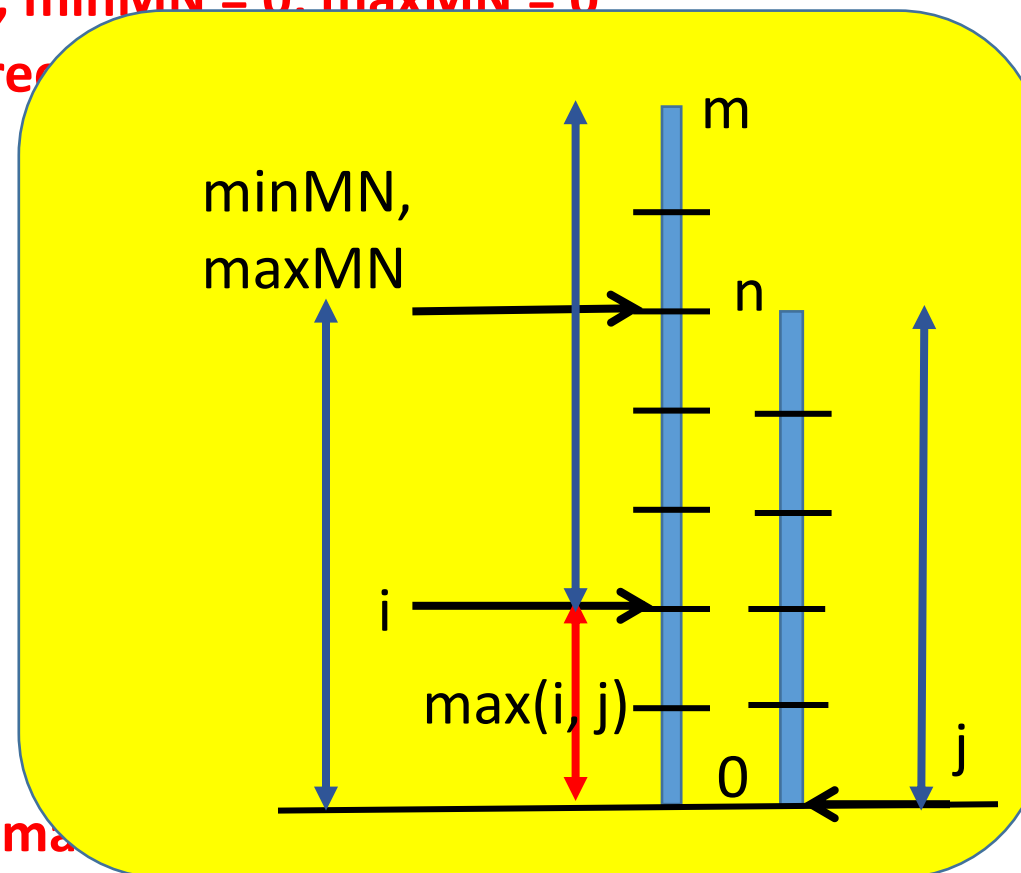
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \min(m, n), \text{maxMN} = \max(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\max(m,n)}, \text{threeRaisedMin} = 3^{\min(m,n)}$**





# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1$ ,  $n \geq 1$ ,  $\text{minMN} = 0$ ,  $\text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1$ ,  $\text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:  $\text{min}(i, j) + \text{minMN} = \text{min}(m, n)$  when both  $i, j$  are  $\geq 0$**

**// When one of  $i$  or  $j$  becomes zero for first time,  $0 + \text{minMN} = \text{min}(m, n)$**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
```

```
{ if ((i >= 1) && (j >= 1)) {
```

```
    minMN++; threeRaisedMin *= 3;
```

```
}
```

```
    maxMN++; twoRaisedMax *= 2;
```

```
}
```

**// POSTCONDITION:  $\text{minMN} = \text{min}(m, n)$ ,  $\text{maxMN} = \text{max}(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\text{max}(m, n)}$ ,  $\text{threeRaisedMin} = 3^{\text{min}(m, n)}$**

# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:  $\text{min}(i, j) + \text{minMN} = m$**

**//**

**// LOOP VARIANT:**

for ( $i = m, j = n; ((i \geq 1) \parallel (j \geq 1)); i--, j--$ )

{ if ( $(i \geq 1) \&\& (j \geq 1)$ ) {

$\text{minMN}++; \text{threeRaisedMin} *= 3;$

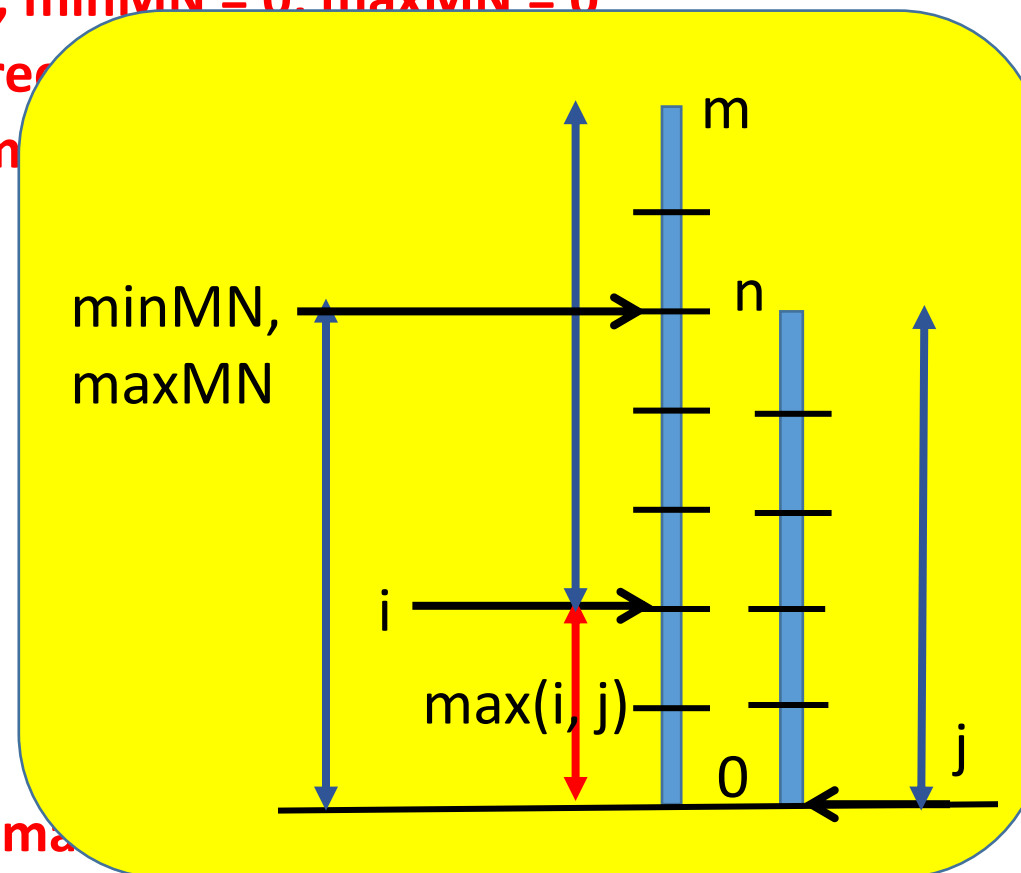
}

$\text{maxMN}++; \text{twoRaisedMax} *= 2;$

}

**// POSTCONDITION:  $\text{minMN} = \text{min}(m, n), \text{maxMN} = \text{max}(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\text{max}(m,n)}, \text{threeRaisedMin} = 3^{\text{min}(m,n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:  $\text{min}(i, j) + \text{minMN} = m$**

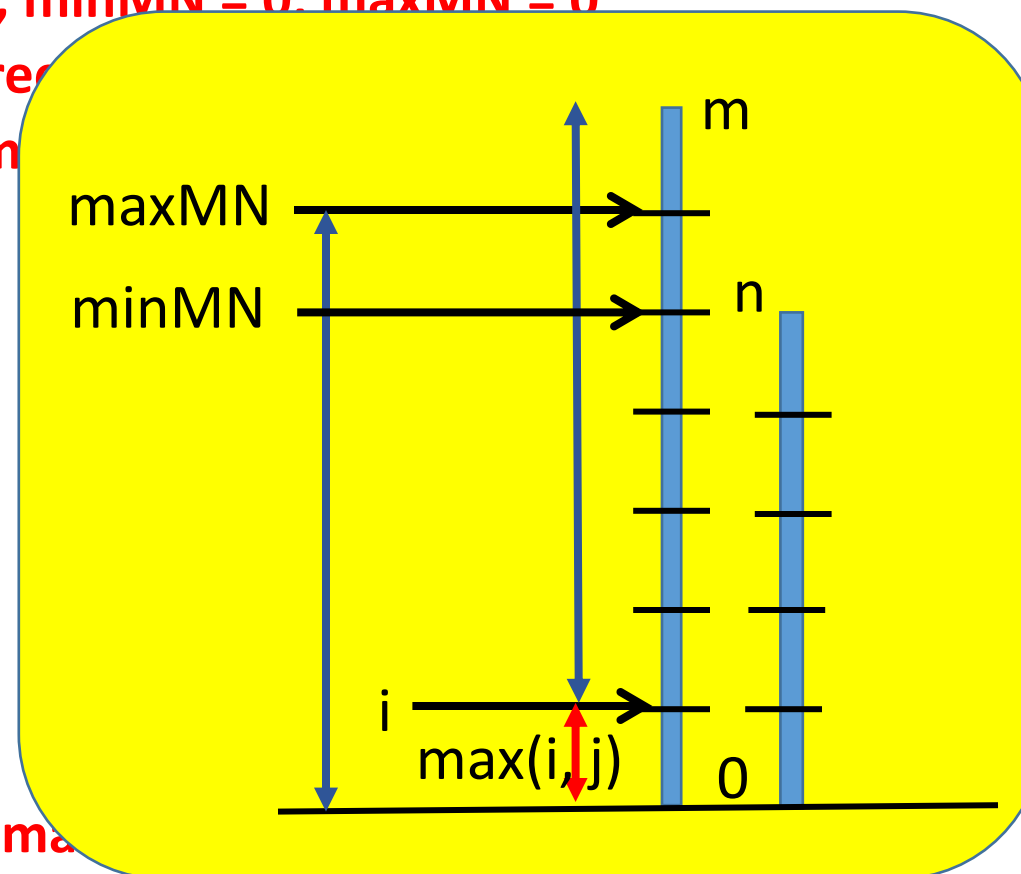
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \text{min}(m, n), \text{maxMN} = \text{max}(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\text{max}(m, n)}, \text{threeRaisedMin} = 3^{\text{min}(m, n)}$**



# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:  $\text{min}(i, j) + \text{minMN} = m$**

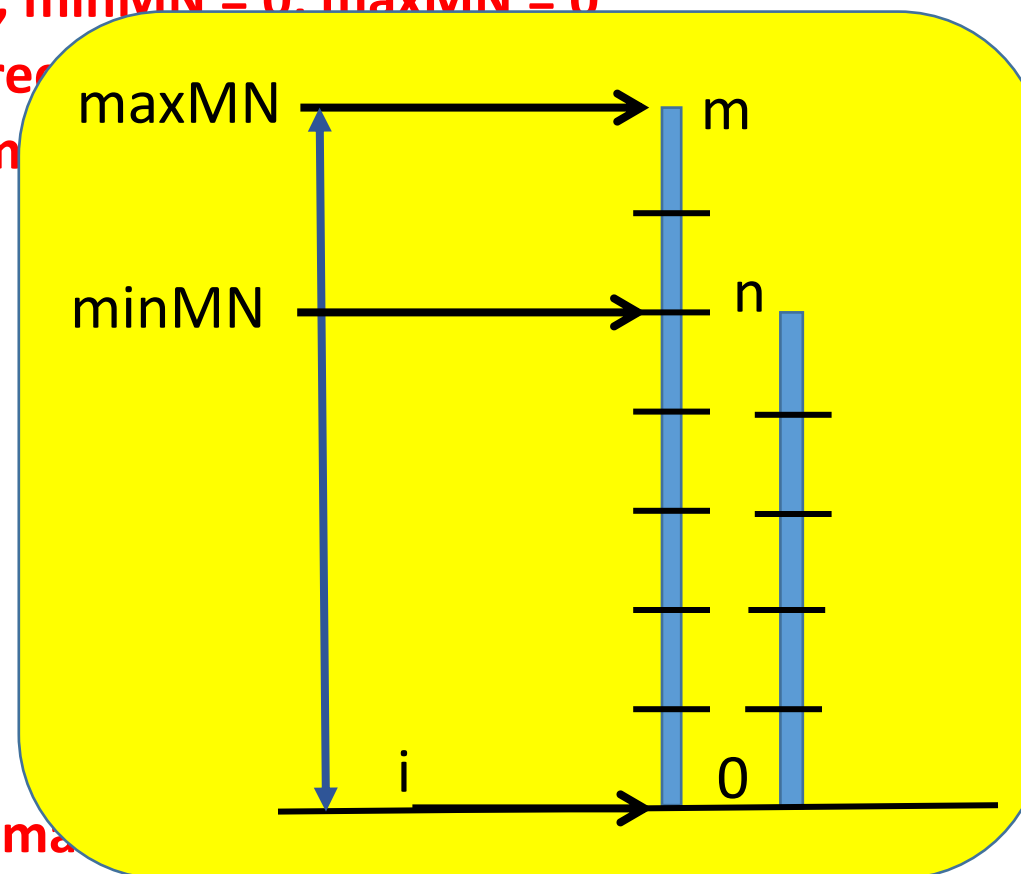
**//**

**// LOOP VARIANT:**

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
{ if ((i >= 1) && (j >= 1)) {
    minMN++; threeRaisedMin *= 3;
  }
  maxMN++; twoRaisedMax *= 2;
}
```

**// POSTCONDITION:  $\text{minMN} = \text{min}(m, n), \text{maxMN} = \text{max}(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\text{max}(m,n)}, \text{threeRaisedMin} = 3^{\text{min}(m,n)}$**



# Reasoning About Loops

```
// PRECONDITION: integers  $m \geq 1$ ,  $n \geq 1$ ,  $\text{minMN} = 0$ ,  $\text{maxMN} = 0$   
//                 $\text{twoRaisedMax} = 1$ ,  $\text{threeRaisedMin} = 1$   
// LOOP INVARIANT:  $\text{min}(i, j) + \text{minMN} = \text{min}(m, n)$  when both  $i, j$  are  $\geq 0$   
//                 $\text{max}(i, j) + \text{maxMN} = \text{max}(m, n)$  when at least one of  $i, j \geq 0$   
//                When the last of  $i$  or  $j$  becomes zero,  $0 + \text{maxMN} = \text{max}(m, n)$ 
```

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)  
{ if ((i >= 1) && (j >= 1)) {  
    minMN++; threeRaisedMin *= 3;  
}  
    maxMN++; twoRaisedMax *= 2;  
}
```

```
// POSTCONDITION:  $\text{minMN} = \text{min}(m, n)$ ,  $\text{maxMN} = \text{max}(m, n)$   
//                 $\text{twoRaisedMax} = 2^{\text{max}(m, n)}$ ,  $\text{threeRaisedMin} = 3^{\text{min}(m, n)}$ 
```

# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:  $\text{max}(0, \text{min}(i, j)) + \text{minMN} = \text{min}(m, n)$**

**//  $\text{max}(i, j) + \text{maxMN} = \text{max}(m, n)$**

**// LOOP VARIANT:**

for ( $i = m, j = n; ((i \geq 1) \parallel (j \geq 1)); i--, j--$ )

{ if ( $(i \geq 1) \&\& (j \geq 1)$ ) {

$\text{minMN}++; \text{threeRaisedMin} *= 3;$

}

$\text{maxMN}++; \text{twoRaisedMax} *= 2;$

}

**// POSTCONDITION:  $\text{minMN} = \text{min}(m, n), \text{maxMN} = \text{max}(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\text{max}(m, n)}, \text{threeRaisedMin} = 3^{\text{min}(m, n)}$**

Taking care of  
“when both  $i, j \geq 0$ ”

# Reasoning About Loops

**// PRECONDITION: integers  $m \geq 1$ ,  $n \geq 1$ ,  $\text{minMN} = 0$ ,  $\text{maxMN} = 0$**

**//  $\text{twoRaisedMax} = 1$ ,  $\text{threeRaisedMin} = 1$**

**// LOOP INVARIANT:  $\text{max}(0, \text{min}(i, j)) + \text{minMN} = \text{min}(m, n)$ ,  $\text{threeRaisedMin} = 3^{\text{minMN}}$**

**//  $\text{max}(i, j) + \text{maxMN} = \text{max}(m, n)$ ,  $\text{twoRaisedMax} = 2^{\text{maxMN}}$**

**// LOOP VARIANT:**

for ( $i = m$ ,  $j = n$ ;  $((i \geq 1) \mid\mid (j \geq 1))$ ;  $i--$ ,  $j--$ )

{ if  $((i \geq 1) \ \&\& \ (j \geq 1))$  {

$\text{minMN}++$ ;  $\text{threeRaisedMin} *= 3$ ;

}

$\text{maxMN}++$ ;  $\text{twoRaisedMax} *= 2$ ;

}

**// POSTCONDITION:  $\text{minMN} = \text{min}(m, n)$ ,  $\text{maxMN} = \text{max}(m, n)$**

**//  $\text{twoRaisedMax} = 2^{\text{max}(m, n)}$ ,  $\text{threeRaisedMin} = 3^{\text{min}(m, n)}$**

# Reasoning About Loops

```
// PRECONDITION: integers  $m \geq 1, n \geq 1, \text{minMN} = 0, \text{maxMN} = 0$ 
```

```
//                 $\text{twoRaisedMax} = 1, \text{threeRaisedMin} = 1$ 
```

```
// LOOP INVARIANT:  $\text{max}(0, \text{min}(i, j)) + \text{minMN}$ 
```

```
//                 $\text{max}(i, j) + \text{maxMN}$ 
```

```
// LOOP VARIANT:  $\text{max}(i, j)$ 
```

```
for (i = m, j = n; ((i >= 1) || (j >= 1)); i--, j--)
```

```
{ if ((i >= 1) && (j >= 1)) {
```

```
    minMN++; threeRaisedMin *= 3;
```

```
}
```

```
    maxMN++; twoRaisedMax *= 2;
```

```
}
```

```
// POSTCONDITION:  $\text{minMN} = \text{min}(m, n), \text{maxMN} = \text{max}(m, n)$ 
```

```
//                 $\text{twoRaisedMax} = 2^{\text{max}(m, n)}, \text{threeRaisedMin} = 3^{\text{min}(m, n)}$ 
```

Non-negative integer-valued expression that monotonically decreases towards 0 in every iteration

Hence, loop terminates



# Reasoning About Loops

```
// PRECONDITION: integers  $m \geq 1$ ,  $n \geq 1$ ,  $\text{minMN} = 0$ ,  $\text{maxMN} = 0$   
//           twoRaisedMax = 1, threeRaisedMin = 1  
// LOOP INVARIANT:  $\max(0, \min(i, j)) + \text{minMN} = \min(m, n)$   
//            $\max(i, j) + \text{maxMN} = \max(m, n)$   
// LOOP VARIANT:  $\max(i, j)$ 
```

Obtaining right loop invariant/variant after design not always easy,  
but crucial to reason about loops

**Best practice: Write pre-conditions, post-conditions, loop invariants, loop variants as comments when programming**

```
// POSTCONDITION:  $\text{minMN} = \min(m, n)$ ,  $\text{maxMN} = \max(m, n)$   
//           twoRaisedMax =  $2^{\max(m, n)}$ , threeRaisedMin =  $3^{\min(m, n)}$ 
```

# Summary

---



- Reasoning about loops in programs
- Pre-conditions, post-conditions, loop invariants and loop variants
- Importance of comments