

# Frequency Count of Expression, Assignment, Conditional, and Iterative Statements (CCDSALG)

Pau Rivera



# ALGORITHMS

## Example

**Input:** a sequence of  $n$  numbers  $\{a_1, a_2, \dots, a_n\}$

**Output:** a permutation (reordering)  $\{a'_1, a'_2, \dots, a'_n\}$  such that  $a'_1 \leq a'_2 \leq \dots \leq a'_n$

**Solution:** ???

# ALGORITHMS

## Example

**Input:** a sequence of  $n$  numbers  $\{a_1, a_2, \dots, a_n\}$

**Output:** a permutation (reordering)  $\{a'_1, a'_2, \dots, a'_n\}$  such that  $a'_1 \leq a'_2 \leq \dots \leq a'_n$

**Solution:** Sorting algorithms

# ALGORITHMS

	 Insertion	 Selection	 Bubble	 Shell	 Merge	 Heap	 Quick	 Quick3
 Random								
 Nearly Sorted								
 Reversed								
 Few Unique								

[Image credits](#)

# Evaluation

*Understanding how  
solutions that solve the  
same problem can differ*



# 4 questions to ask when evaluating a solution

Does it provide a satisfactory way for  
the target audience to use it?

Is it  
USABLE?

Is it simple yet effective?

Is it  
ELEGANT?

Does it use resources  
reasonably?

Is it  
EFFICIENT?

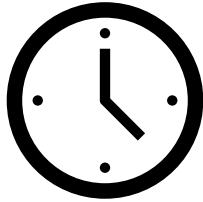
Does your solution actually solve  
the problem?

Is it  
CORRECT?



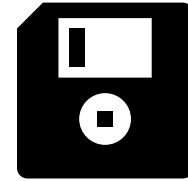
# Efficiency

- Computers utilize **resources** (memory, computation)



**Time**

Duration of an algorithm's run  
time from start to end

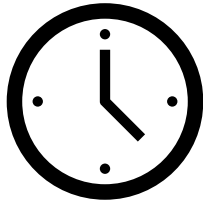


**Space**

Amount of memory needed to  
process

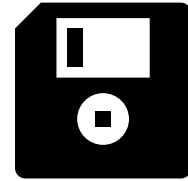
# Efficiency

- Computers utilize **resources** (memory, computation)



**Time**

What parts of the algorithm affects the runtime?

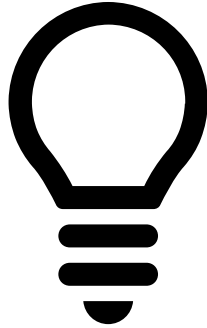


**Space**

How does the choice of data structure affect the runtime?



# ANALYSIS METHODS



## A **priori** Analysis

Obtains a function bounding the time complexity from mathematical facts.



## A **posteriori** Analysis

Study the exact time and space required for execution using actual experiments

# Frequency Count

- Number of statements or steps needed by the algorithm to finish
- Used to approximate the running time with the assumption that 1 statement or step is equivalent to 1 time unit
- Denoted as  $T(n)$

# Example

<pre>total = float(input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  change = tendered - total  print("Change: ", change)</pre>	
---	--

# Example

```
total = float(Input("Grand Total: "))  
tendered = float(input("Amount Tendered: "))  
  
change = tendered - total  
  
print("Change: ", change)
```

1

# Example

```
total = float(Input("Grand Total: "))
```

```
tendered = float(input("Amount Tendered: "))
```

```
change = tendered - total
```

```
print("Change: ", change)
```

1

1

# Example

```
total = float(Input("Grand Total: "))  
tendered = float(input("Amount Tendered: "))  
  
change = tendered - total  
  
print("Change: ", change)
```

1

1

1

# Example

<code>total = float(Input("Grand Total: "))</code>	1
<code>tendered = float(input("Amount Tendered: "))</code>	1
<code>change = tendered - total</code>	1
<code>print("Change: ", change)</code>	1

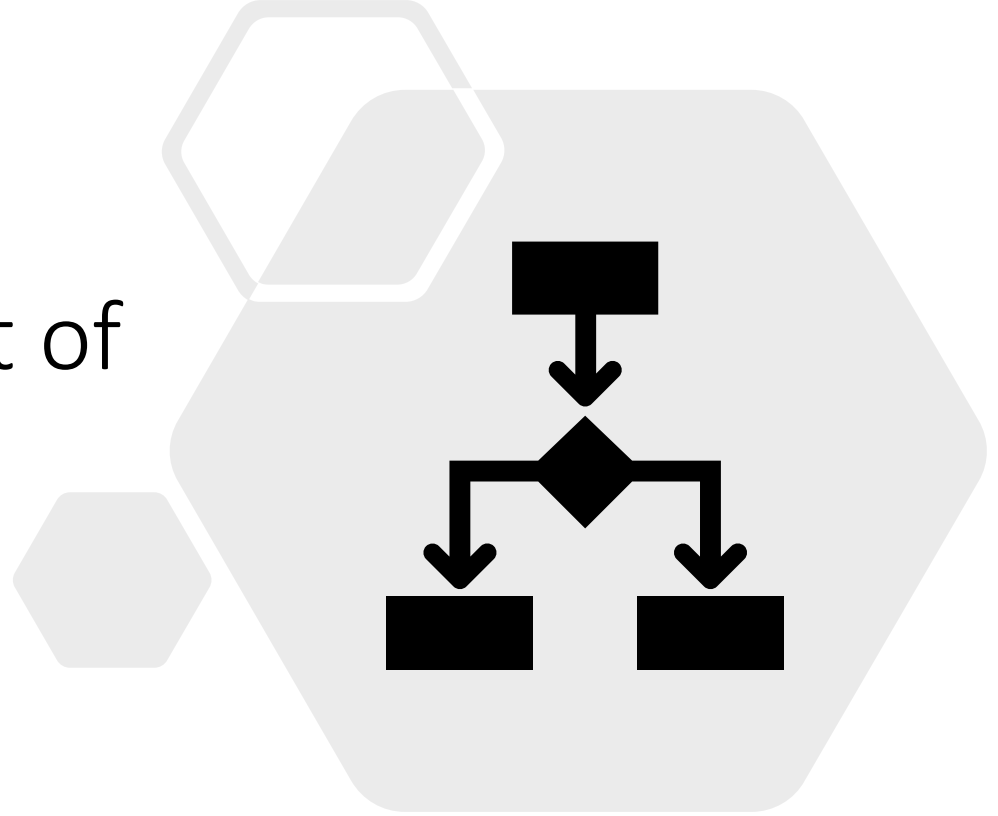
# Example

<code>total = float(Input("Grand Total: "))</code>	1
<code>tendered = float(input("Amount Tendered: "))</code>	1
<code>change = tendered - total</code>	1
<code>print("Change: ", change)</code>	1

Frequency Count	4
Growth Rate	O(1)



# Frequency Count of Conditional Statements



# Example for Conditional Statements

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
else if tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

## Example (Case 1: tendered < total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

## Example (Case 2: tendered = total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

## Example (Case 3: tendered > total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

# Frequency Count of Conditional Statements

WE ASSUME THE WORST CASE

- **Worst-case** complexity = **maximum** number of steps
- Best-case complexity = minimum number of steps
- Average-case complexity = average number of steps

Case 1: tendered < total

## Example (Case: tendered < total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1



## Example (Case: tendered < total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1  
1

## Example (Case: tendered < total)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	
<pre>elif tendered == total:</pre>	
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: tendered < total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	<pre>1 1 1</pre>
--	------------------

## Example (Case: tendered < total)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	1
<pre>elif tendered == total:</pre>	
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: tendered < total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	1 1  1 1
Frequency Count	4

Case 2: tendered = total

## Example (Case: tendered = total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1

## Example (Case: tendered = total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1  
1



## Example (Case: tendered = total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1

1

1

## Example (Case: tendered = total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	<pre>1 1 1</pre>
--	------------------

## Example (Case: tendered = total)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	
<pre>elif tendered == total:</pre>	1
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: **tendered = total**)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	
<pre>elif <b>tendered == total</b>:</pre>	1
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: tendered = total)

<code>total = float(Input("Grand Total: "))</code>	1
<code>tendered = float(input("Amount Tendered: "))</code>	1
<code>if tendered &lt; total:</code>	1
<code>print("It's less than the Grand Total.")</code>	
<code>elif tendered == total:</code>	1
<code>print("You have given the exact amount")</code>	1
<code>else:</code>	
<code>change = tendered - total</code>	
<code>print("Change: ", change)</code>	

## Example (Case: tendered = total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	1 1  1  1 1   
Frequency Count	5

Case 3: tendered > total

## Example (Case: tendered > total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1



## Example (Case: tendered > total)

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

1  
1

## Example (Case: tendered > total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	<pre>1 1 1</pre>
--	------------------

## Example (Case: tendered > total)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	
<pre>elif tendered == total:</pre>	
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: tendered > total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	<pre>1 1 1</pre>
--	------------------

## Example (Case: tendered > total)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	
<pre>elif tendered == total:</pre>	1
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: tendered > total)

<pre>total = float(Input("Grand Total: ")) tendered = float(input("Amount Tendered: "))  if tendered &lt; total:     print("It's less than the Grand Total.") elif tendered == total:     print("You have given the exact amount") else:     change = tendered - total     print("Change: ", change)</pre>	<pre>1 1 1 1</pre>
--	--------------------

## Example (Case: tendered > total)

<pre>total = float(Input("Grand Total: "))</pre>	1
<pre>tendered = float(input("Amount Tendered: "))</pre>	1
<pre>if tendered &lt; total:</pre>	1
<pre>    print("It's less than the Grand Total.")</pre>	
<pre>elif tendered == total:</pre>	1
<pre>    print("You have given the exact amount")</pre>	
<pre>else:</pre>	
<pre>    change = tendered - total</pre>	
<pre>    print("Change: ", change)</pre>	

## Example (Case: tendered > total)

<code>total = float(Input("Grand Total: "))</code>	1
<code>tendered = float(input("Amount Tendered: "))</code>	1
<code>if tendered &lt; total:</code>	1
<code>print("It's less than the Grand Total.")</code>	
<code>elif tendered == total:</code>	1
<code>print("You have given the exact amount")</code>	
<code>else:</code>	
<code>change = tendered - total</code>	1
<code>print("Change: ", change)</code>	



## Example (Case: tendered > total)

<code>total = float(Input("Grand Total: "))</code>	1
<code>tendered = float(input("Amount Tendered: "))</code>	1
<code>if tendered &lt; total:</code>	1
<code>print("It's less than the Grand Total.")</code>	
<code>elif tendered == total:</code>	1
<code>print("You have given the exact amount")</code>	
<code>else:</code>	
<code>change = tendered - total</code>	1
<code>print("Change: ", change)</code>	1

## Example (Case: tendered > total)

total = float(Input("Grand Total: "))	1
tendered = float(input("Amount Tendered: "))	1
if tendered < total:	1
print("It's less than the Grand Total.")	
elif tendered == total:	1
print("You have given the exact amount")	
else:	
change = tendered - total	1
print("Change: ", change)	1
Frequency Count	6

# Example for Conditional Statements

Case	Frequency Count	Growth Rate
tendered < total	4	$O(1)$
tendered = total	5	$O(1)$
tendered > total	6	$O(1)$

Maximum frequency count: 6

# Example for Conditional Statements

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
elif tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

Frequency Count	6
-----------------	---

# Frequency Count of Conditional Statements

```
if <condition>  
    <S1>  
else  
    <S2>
```

$$Total = 1 + \max(count(< S1 >), count(< S2 >))$$

# Example for Conditional Statements

```
total = float(Input("Grand Total: "))
tendered = float(input("Amount Tendered: "))

if tendered < total:
    print("It's less than the Grand Total.")
else if tendered == total:
    print("You have given the exact amount")
else:
    change = tendered - total
    print("Change: ", change)
```

Frequency Count	
-----------------	--

# Frequency Count of Iterative Statements



# Iterative Statements

Iterations/loops are *iterative*, they repeat a specific number of times. To identify the frequency count of an iterator, we have to trace the looping mechanism used.

The types of loops are:

- While
- Do-While
- For



# Iterative Statements

Loops generally follow a structure of:

- Check the **condition**
  - If the condition is false, **exit**
  - Else, **execute** the body

The While loop does exactly this!

**while** **condition**  
**body**

# WHILE LOOP

```
[1] x = 4  
[2] while x < 6  
[3]     x = x + 1
```

# WHILE LOOP

```
[1] x = 4  
[2] while x < 6  
[3]     x = x + 1
```

```
[1] I  
[2]  
[3]
```

Line 1 will be executed only 1 time.

# WHILE LOOP

```
[1] x = 4  
[2] while x < 6  
[3]     x = x + 1
```

```
[1] I  
[2] I  
[3] I
```

Value of x	Condition	Enter Loop?
4	< 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

# WHILE LOOP

```
[1] x = 4  
[2] while x < 6  
[3]     x = x + 1
```

```
[1] I  
[2] II  
[3] II
```

Value of x	Condition	Enter Loop?
5	< 6	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

# WHILE LOOP

```
[1] x = 4  
[2] while x < 6  
[3]     x = x + 1
```

```
[1] I  
[2] III  
[3] II
```

Value of x	Condition	Enter Loop?
6	< 6	No

Add 1 to the tally of the loop condition only, since the loop checks the condition before it exits.

# WHILE LOOP

```
[1] x = 4  
[2] while x < 6  
[3]     x = x + 1
```

```
[1] I  
[2] III      =      3  
[3] II       =      2
```

**Value of x**

**6**

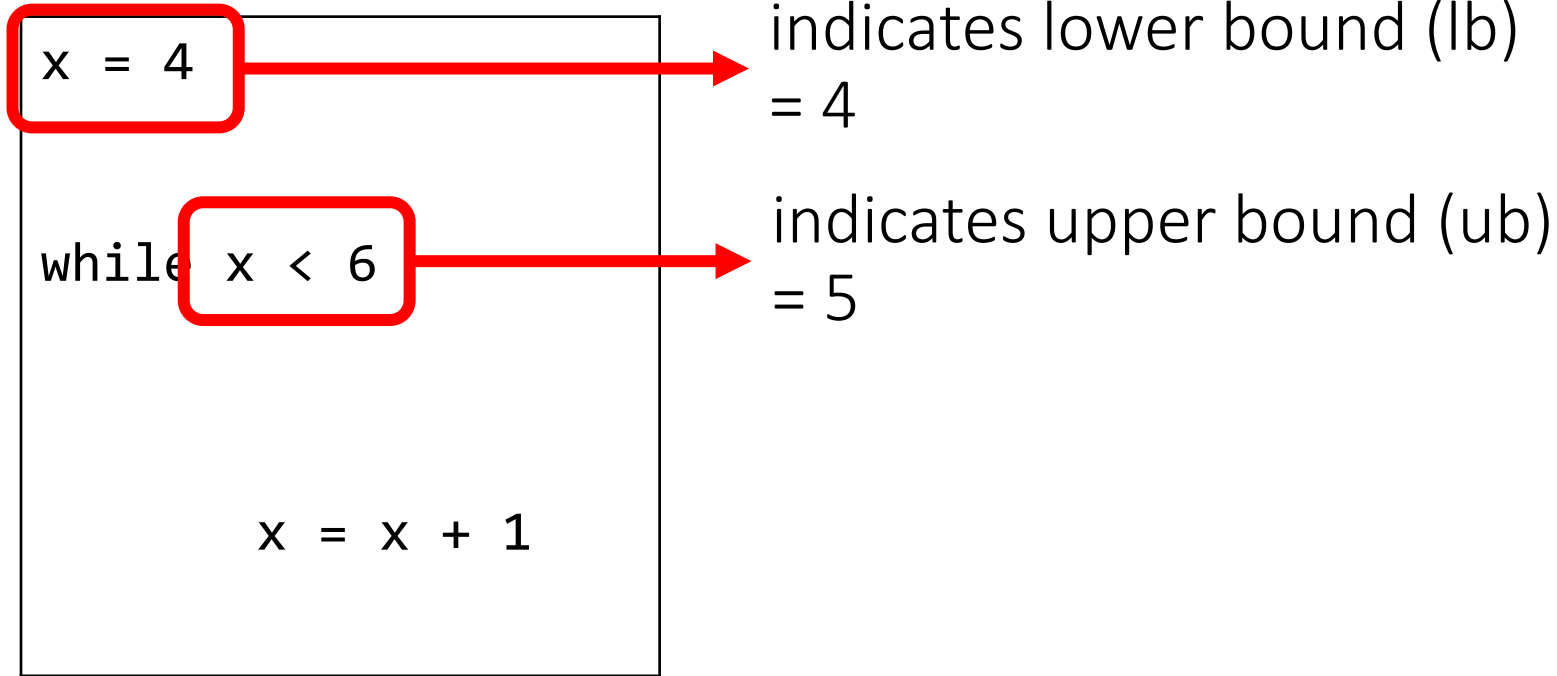
**Condition**

**< 6**

**Enter Loop?**

**No**

How can we get the number of times it looped without manually counting it?





How can we get the number of times it looped without manually counting it?

<pre>x = 4  while x &lt; 6      x = x + 1</pre>	$1$ $\sum_{x=4}^5 1 + 1 = (5 - 4 + 1) + 1 = 3$ $\sum_{x=4}^5 1 = (5 - 4 + 1) = 2$
---	---

Frequency Count

$$1 + 3 + 2 = 6$$

# Iterative Statements

Loops generally follow a structure of:

- Check the **condition**
  - If the condition is false, **exit**
  - Else, **execute** the body

The Do-While loop is very similar,

do  
    body  
while condition

# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

```
[1] I  
[2]  
[3]  
[4]
```

Line 1 will be executed only 1 time.

# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

```
[1] I  
[2]  
[3] I  
[4]
```

Execute the statement inside the loop without checking the condition.

**Value of x**

**2**

# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

```
[1] I  
[2]  
[3] I  
[4] I
```

Check the loop condition.

Value of x	Condition	Enter Loop?
2	< 3	Yes

# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

```
[1] I  
[2]  
[3] II  
[4] I
```

Since the condition is true, enter the loop again.

Value of x	Condition	Enter Loop?
3	< 3	Yes

# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

```
[1] I  
[2]  
[3] II  
[4] II
```

Check the loop condition. Since it is false, exit the loop.

Value of x	Condition	Enter Loop?
3	< 3	No



# DO WHILE LOOP

```
[1] x = 1  
[2] do  
[3]     x = x + 1  
[4] while x < 3
```

```
[1] I  
[2]  
[3] II      =      2  
[4] II      =      2
```

**Value of x**

**3**

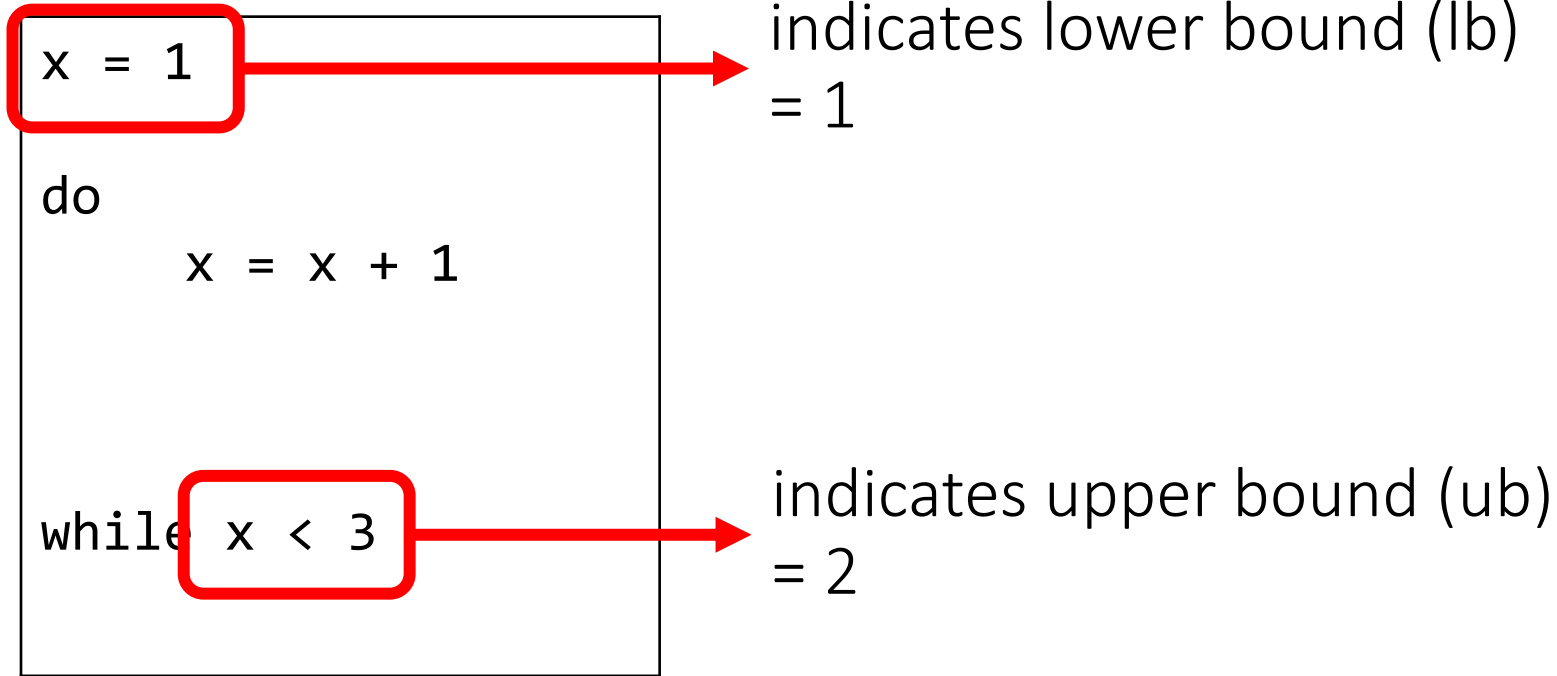
**Condition**

**< 3**

**Enter Loop?**

**No**

How can we get the number of times it looped without manually counting it?



How can we get the number of times it looped without manually counting it?

<pre>x = 1  do     x = x + 1  while x &lt; 3</pre>	$1$ $\sum_{x=1}^2 1 = (2 - 1 + 1) = 2$ $\sum_{x=1}^2 1 = (2 - 1 + 1) = 2$
--	---

Frequency Count

$$1 + 2 + 2 = 5$$

# Iterative Statements

Loops generally follow a structure of:

- Check the **condition**
  - If the condition is false, **exit**
  - Else, **execute** the body

The For loop is a bit more nuanced..

for iterator  
body

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] I  
[2] I
```

Value of i	Condition	Enter Loop?
0	$\leq 4$	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] II  
[2] II
```

Value of i	Condition	Enter Loop?
1	$\leq 4$	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] III  
[2] III
```

Value of i	Condition	Enter Loop?
2	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.



# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] I I I I  
[2] I I I I
```

Value of i	Condition	Enter Loop?
3	$\leq 4$	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] I I I I I  
[2] I I I I I
```

Value of i	Condition	Enter Loop?
4	<= 4	Yes

Add 1 to the tally of both the loop condition and the statements inside the loop, since the condition is true.

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] I I I I I I  
[2] I I I I I
```

Value of i	Condition	Enter Loop?
5	$\leq 4$	No

Add 1 to the tally of the loop condition only, since the loop checks the condition before it exits.

# FOR LOOP

```
[1] for i = 0 to 4  
[2]     print(i)
```

```
[1] IIIIII = 6  
[2] IIIII = 5
```

**Value of i**

**5**

**Condition**

**$\leq 4$**

**Enter Loop?**

**No**

How can we get the number of times it looped without manually counting it?

```
for i = 0 to 4  
  
    print(i)
```

indicates lower bound (lb)  
= 0

indicates upper bound (ub)  
= 4

How can we get the number of times it looped without manually counting it?

```
for i = 0 to 4
```

```
    print(i)
```

$$\sum_{i=0}^4 1 + 1 = (4 - 0 + 1) + 1 = 6$$

$$\sum_{i=0}^4 1 = (4 - 0 + 1) = 5$$

Frequency Count

$$6 + 5 = 11$$

# Example

```
x = 1
```

```
do
```

```
    y = y + 1
```

```
    x = x + 1
```

```
while x != n - 1
```

Frequency Count	
-----------------	--



Questions? 😊

