Welcome back!

Today:

- short lecture; don't worry about taking notes if it feels fast
- group activity

Tomorrow:

- continuation of group activity

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array

input: array, way to compare elements

output: sorted array

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array
    input: array, way to compare elements
    output: sorted array

ex: compute the factorial of a positive integer
    input: a positive integer n
    output: n!

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array
    input: array, way to compare elements
    output: sorted array

ex: compute the factorial of a positive integer
    input: a positive integer n
    output: n!

note that we need the tools of discrete math to define these inputs and outputs precisely

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array
    input: array, way to compare elements
    output: sorted array

ex: compute the factorial of a positive integer
    input: a positive integer n
    output: n!

note that we need the tools of discrete math to define these inputs and outputs precisely

A **solution** is some method of taking in an arbitrary input and computing an output with the desired properties defined by the problem

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array
    input: array, way to compare elements
    output: sorted array

ex: compute the factorial of a positive integer
    input: a positive integer n
    output: n!

note that we need the tools of discrete math to define these inputs and outputs precisely

A **solution** is some method of taking in an arbitrary input and computing an output with the desired properties defined by the problem; typically an **algorithm**, a sequence of steps you can perform to get from input to output.

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array
    input: array, way to compare elements
    output: sorted array

note that we need the tools of discrete math to define these inputs and outputs precisely

ex: compute the factorial of a positive integer
    input: a positive integer n
    output: n!

A **solution** is some method of taking in an arbitrary input and computing an output with the desired properties defined by the problem; typically an **algorithm**, a sequence of steps you can perform to get from input to output.

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

A **problem** for a computer must be defined precisely and unambiguously by its input and desired output.

ex: sort an array
    input: array, way to compare elements
    output: sorted array

note that we need the tools of discrete math to define these inputs and outputs precisely

ex: compute the factorial of a positive integer
    input: a positive integer n
    output: n!

A **solution** is some method of taking in an arbitrary input and computing an output with the desired properties defined by the problem; typically an **algorithm**, a sequence of steps you can perform to get from input to output.

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

*algorithms are represented using **pseudocode**, a mix of precise/unambiguous notation and words*

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

1) Does the algorithm work?

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

1) Does the algorithm work? *future classes*

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

1) Does the algorithm work? *future classes*

2) Does the algorithm work efficiently? *this class: runtime*

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

1) Does the algorithm work? *future classes*

2) Does the algorithm work efficiently? *this class: runtime*

How to measure runtime?

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

1) Does the algorithm work? *future classes*

2) Does the algorithm work efficiently? *this class: runtime*

How to measure runtime?

Idea #1: implement the algorithm, run it, time it...

- depends on software, hardware, operating system, etc.
- implementation takes time and is error-prone
- how do we choose which inputs to run it on?

```
sort(A):
    let S = the set of all permutations of A
    for x in S:
        if x is sorted:
            return x
```

1) Does the algorithm work? *future classes*

2) Does the algorithm work efficiently? *this class: runtime*

How to measure runtime?

Idea #1: implement the algorithm, run it, time it...

- depends on software, hardware, operating system, etc.
- implementation takes time and is error-prone
- how do we choose which inputs to run it on?

Idea #2: find a function that expresses runtime in terms of input size

- runtime: number of primitive operations (arithmetic operations, logical operations, variable retrieval, variable assignment, etc.)

Algorithm 1

```
sum = 0
for i = 1 to n:
        sum = sum + 1
for i = 1 to n:
        sum = sum + 1
for i = 1 to n:
        sum = sum + 1
```

Algorithm 1

```
sum = 0
for i = 1 to n:
        sum = sum + 1
for i = 1 to n:
        sum = sum + 1
for i = 1 to n:
        sum = sum + 1
```

variable assignment

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1

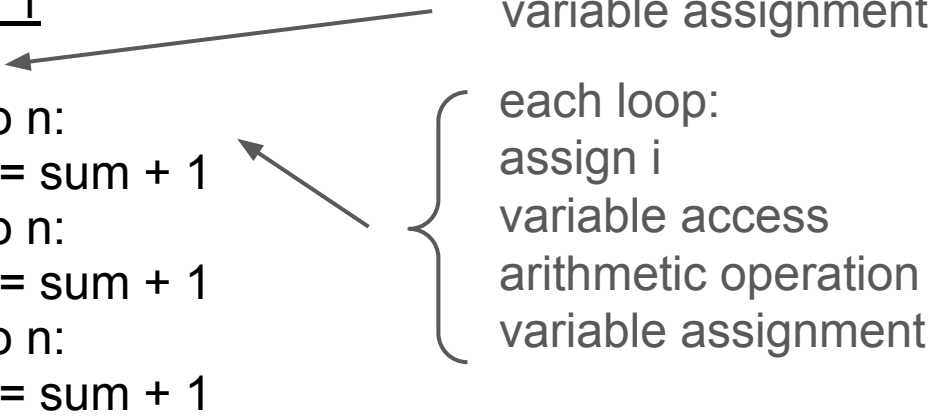
```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

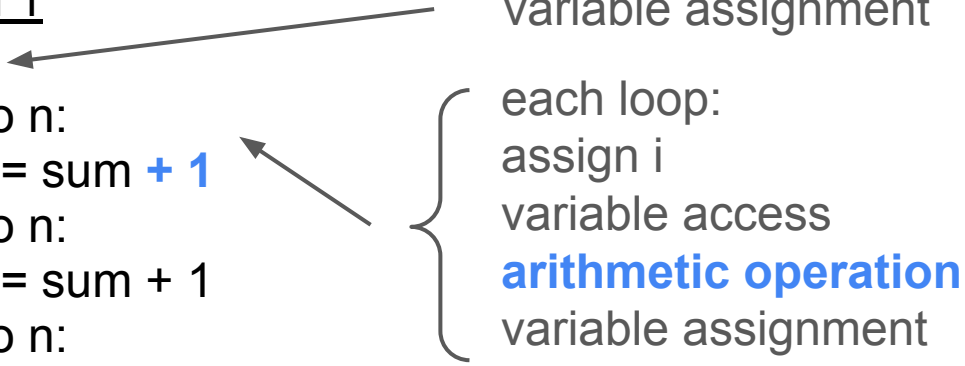
Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```



Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

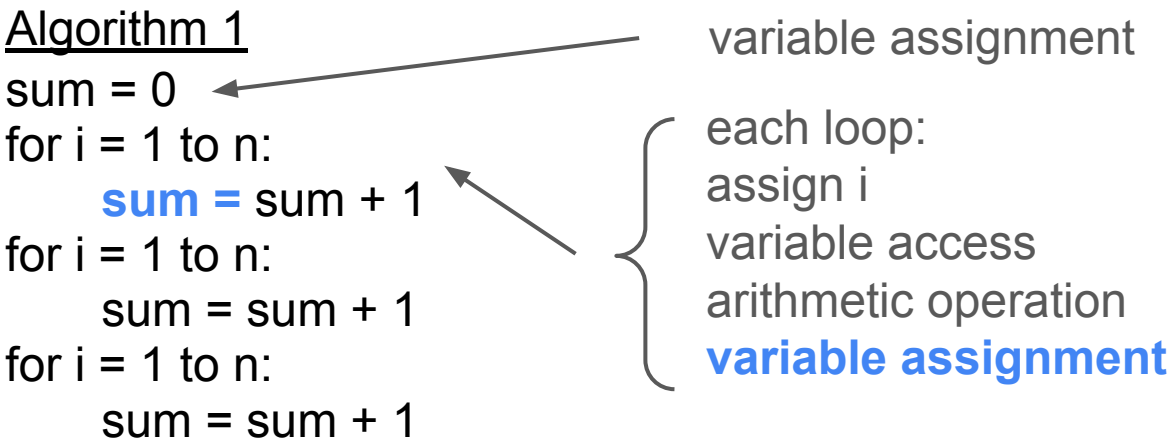


Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 1 to n:
sum = sum + 1
for i = 1 to n:
sum = sum + 1
for i = 1 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

1+

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

1+3

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

1+3*n

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

1+3*n*4

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

$1+3*n*4 = 12n+1$

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

Algorithm 1

```
sum = 0
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
for i = 1 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

Algorithm 2

```
sum = 0
if n < 100
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

Algorithm 1

```
sum = 0
for i = 0 to n:
    sum = sum + 1
for i = 0 to n:
    sum = sum + 1
for i = 0 to n:
    sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$f_1(n) = 12n+1$

variable assignment

variable access
logical operation

Algorithm 2

```
sum = 0
if n < 100:
    for i = 1 to n:
        for j = 1 to n:
            sum = sum + n/3
else:
    sum = 3n
```

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$f_1(n) = 12n+1$

variable assignment

variable access
logical operation

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

<u>Algorithm 1</u>

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$f_1(n) = 12n+1$

variable assignment

variable access
logical operation

each outer loop:
variable assignment
inner loop

<u>Algorithm 2</u>

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

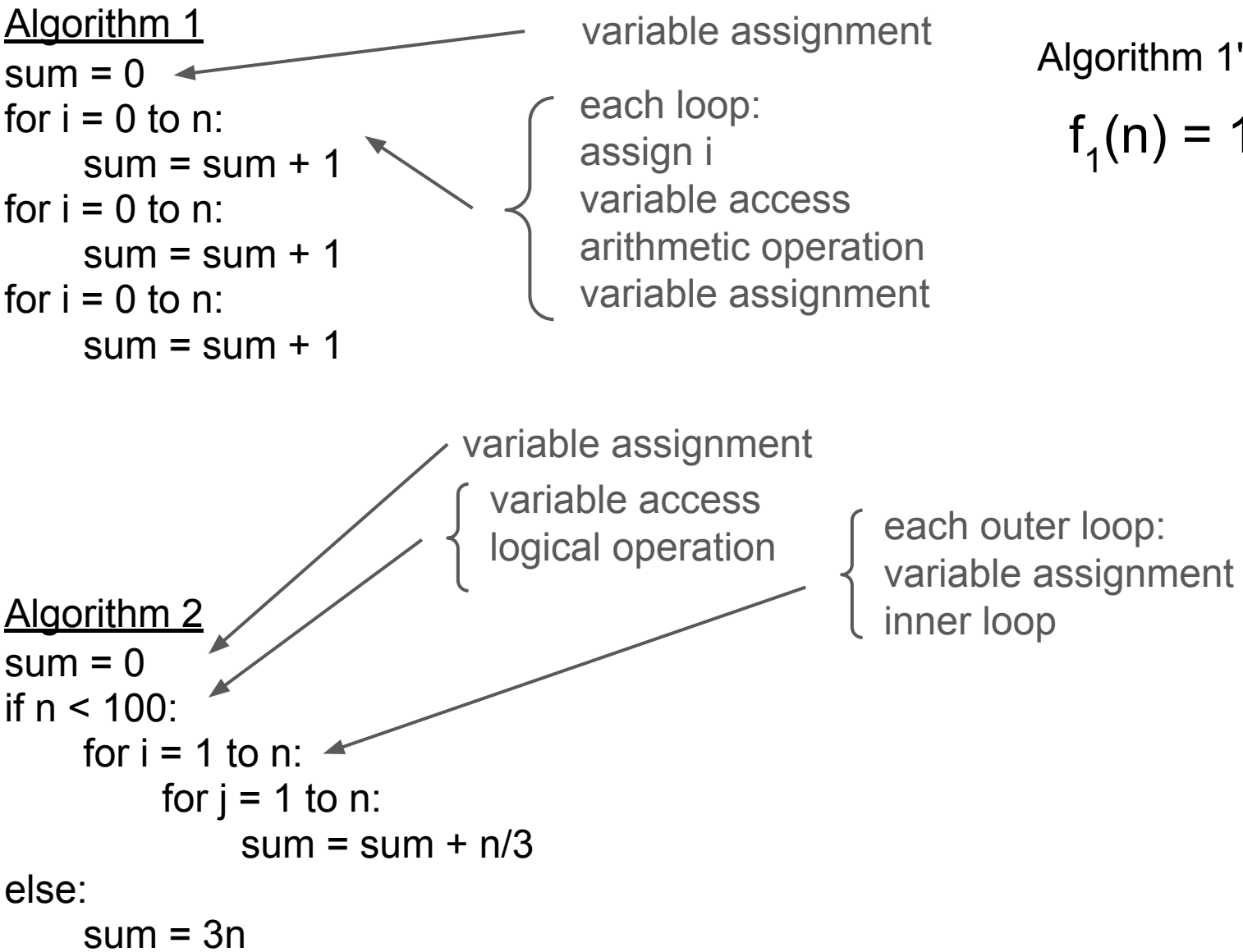
variable assignment

variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

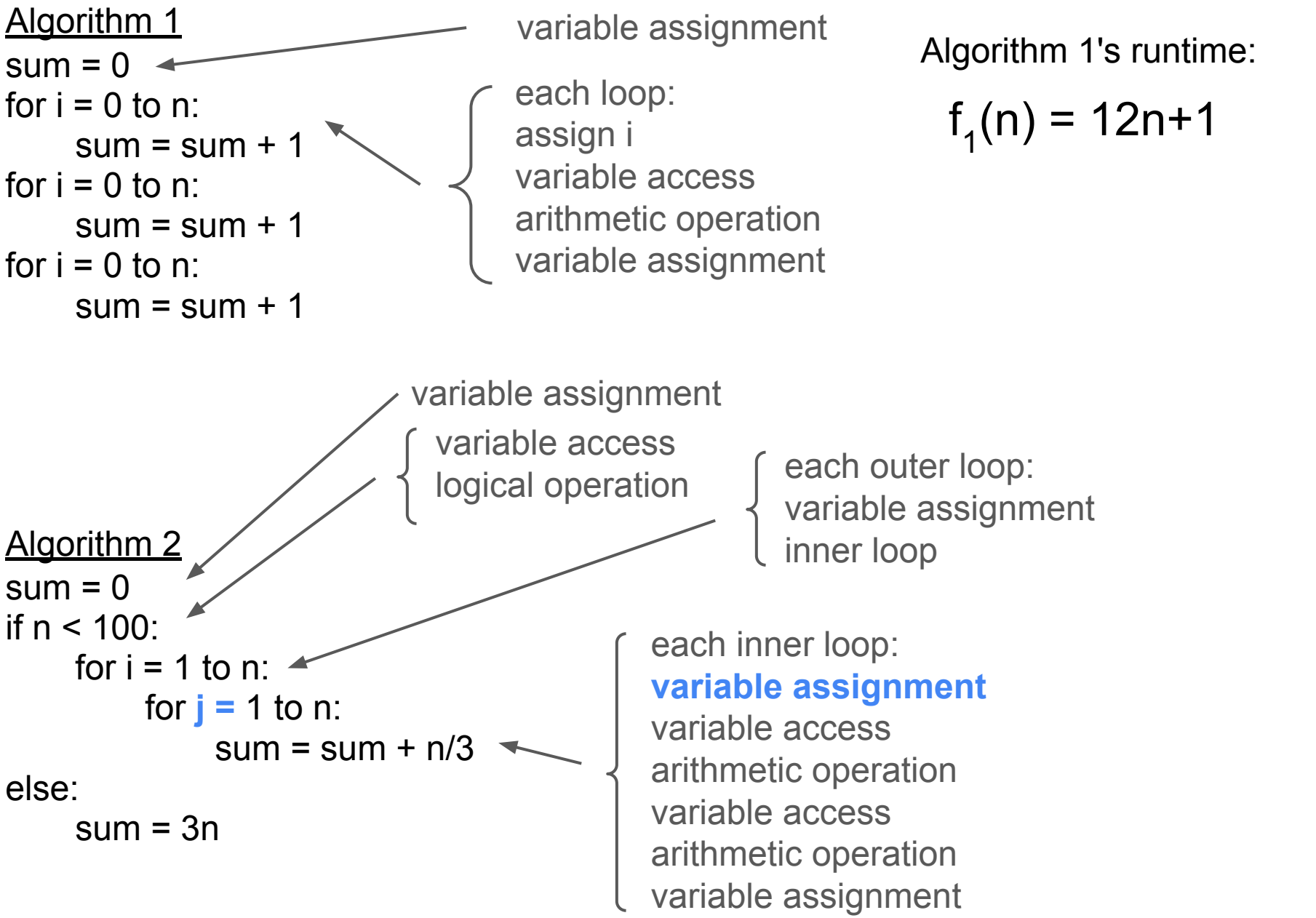
variable access
logical operation

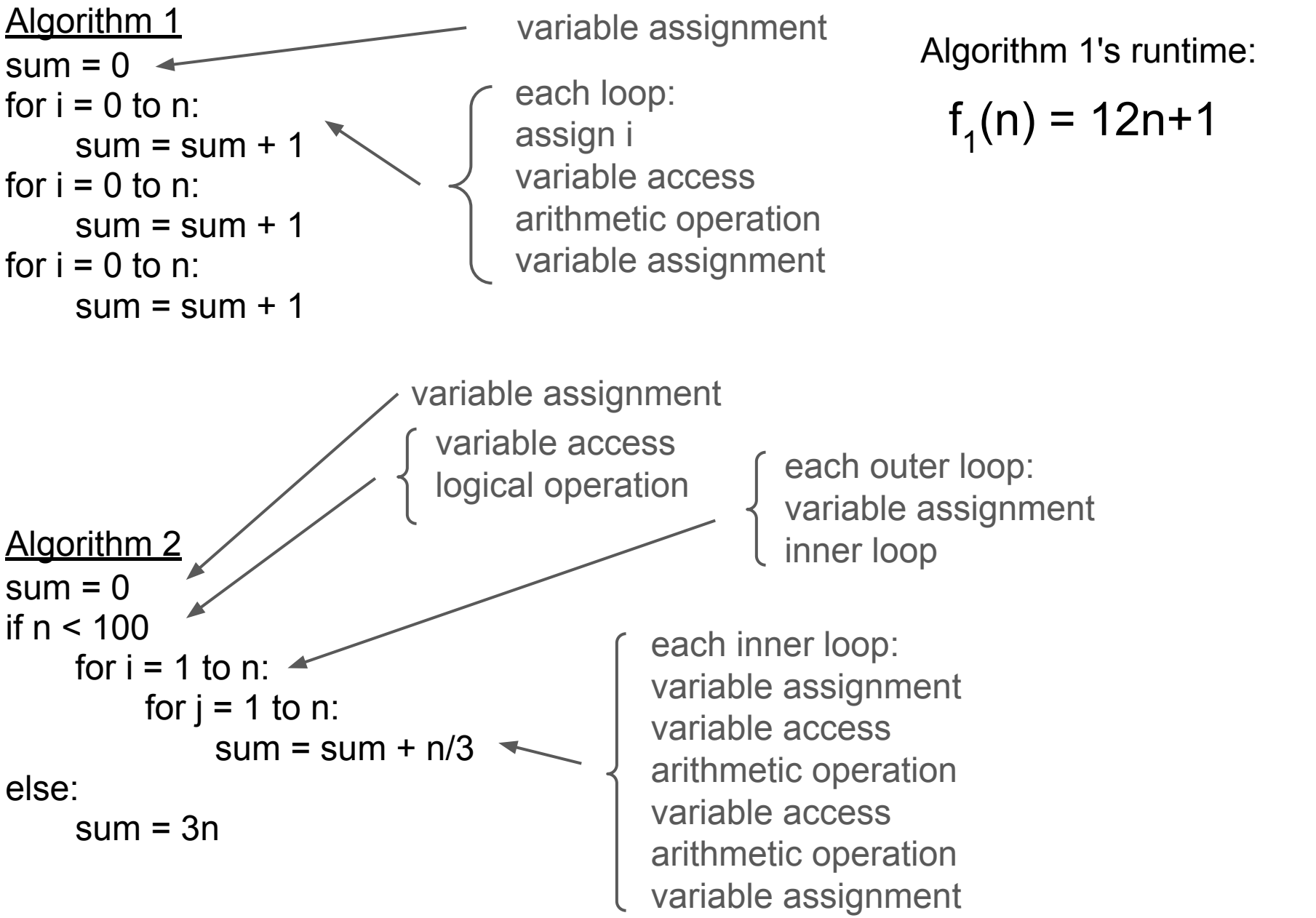
each outer loop:
variable assignment
inner loop

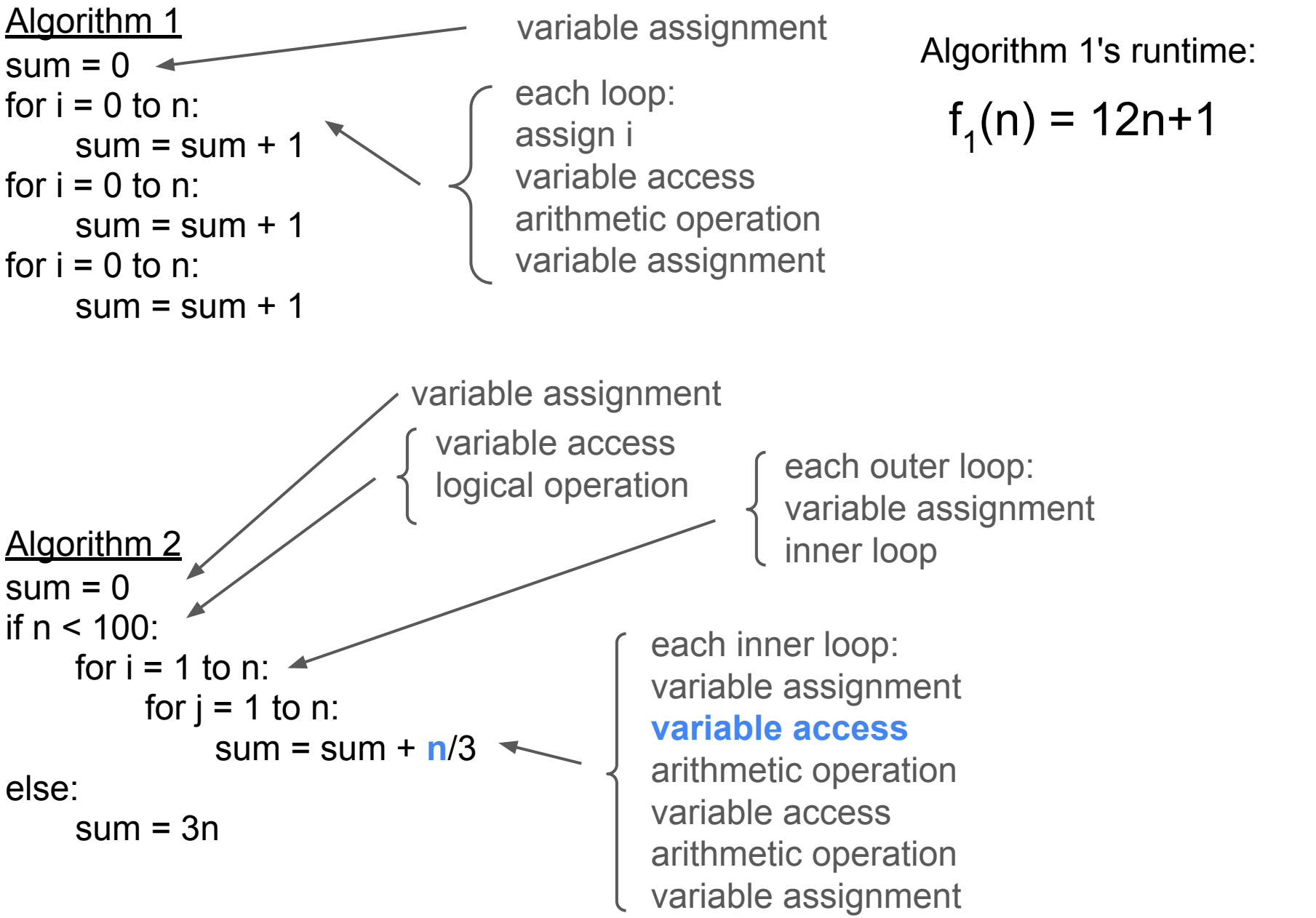
Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

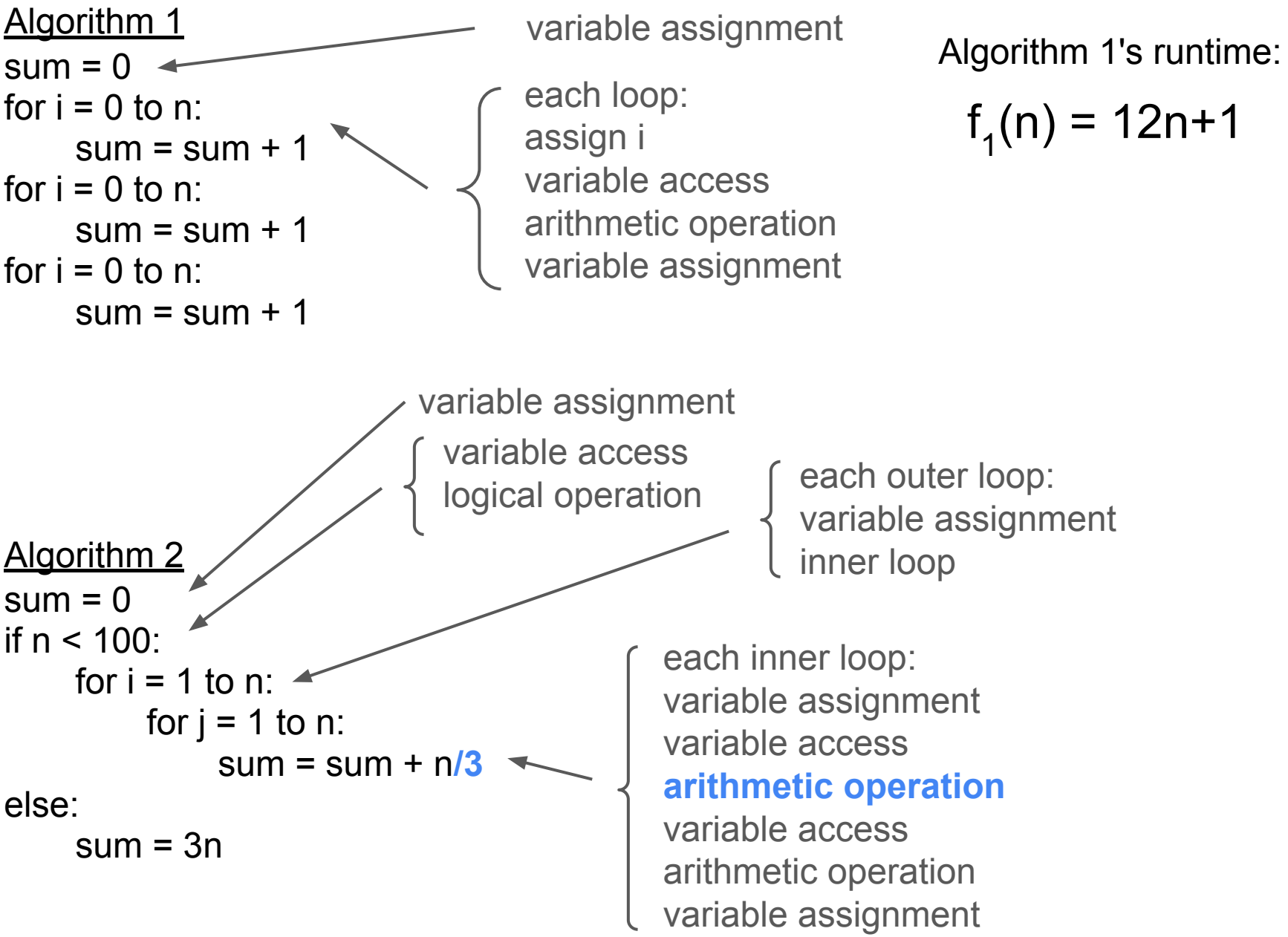
variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

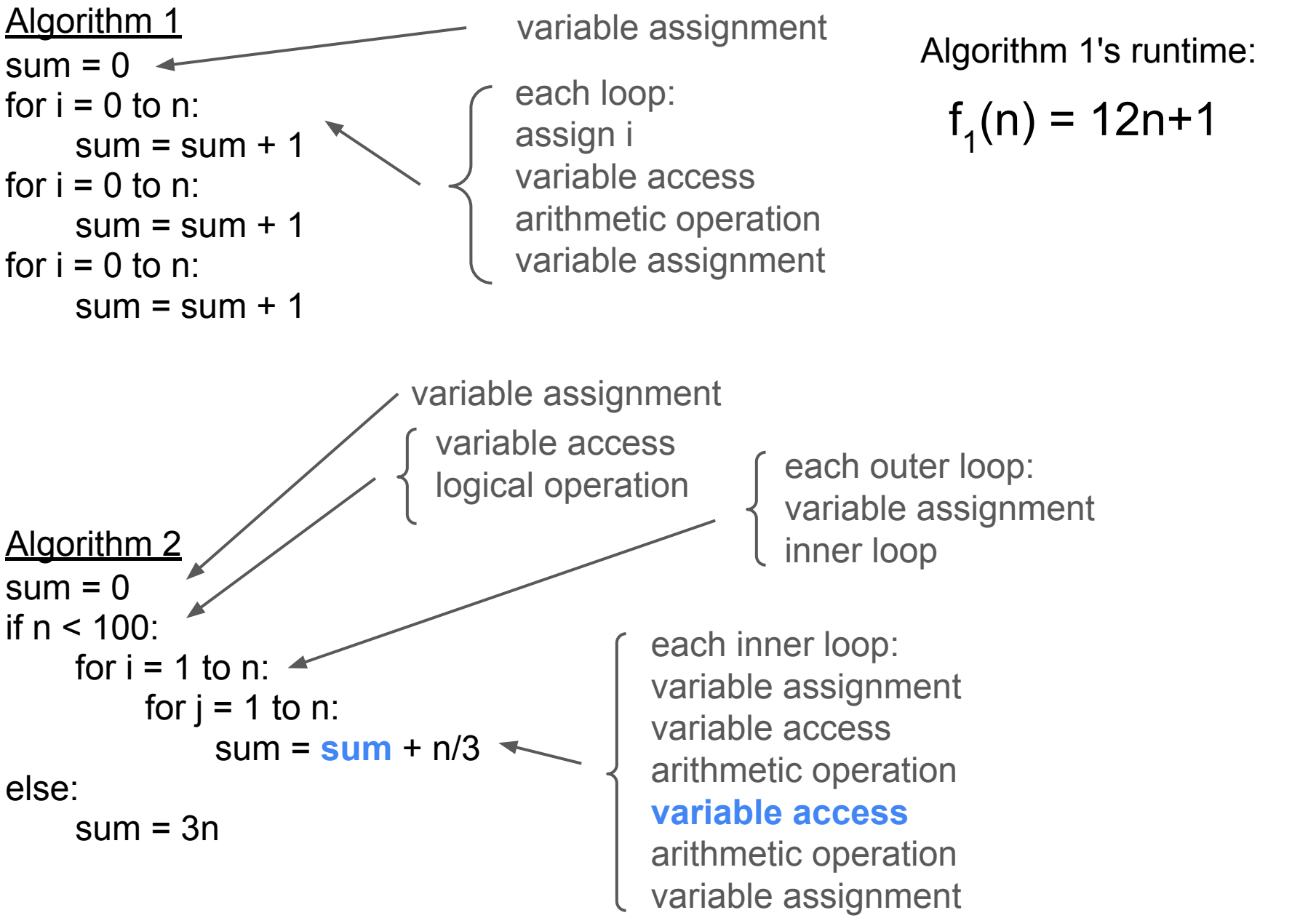
variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

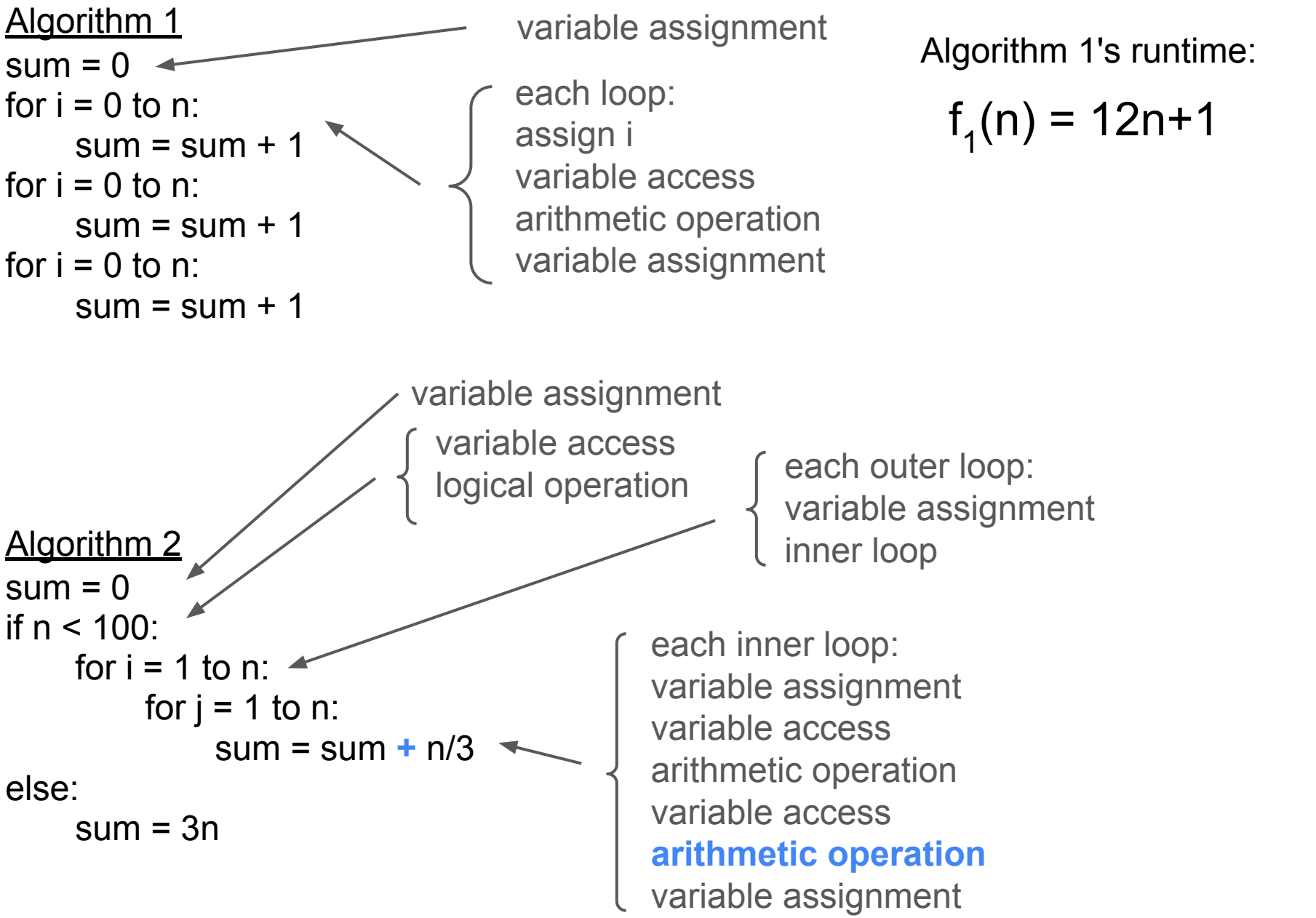
variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

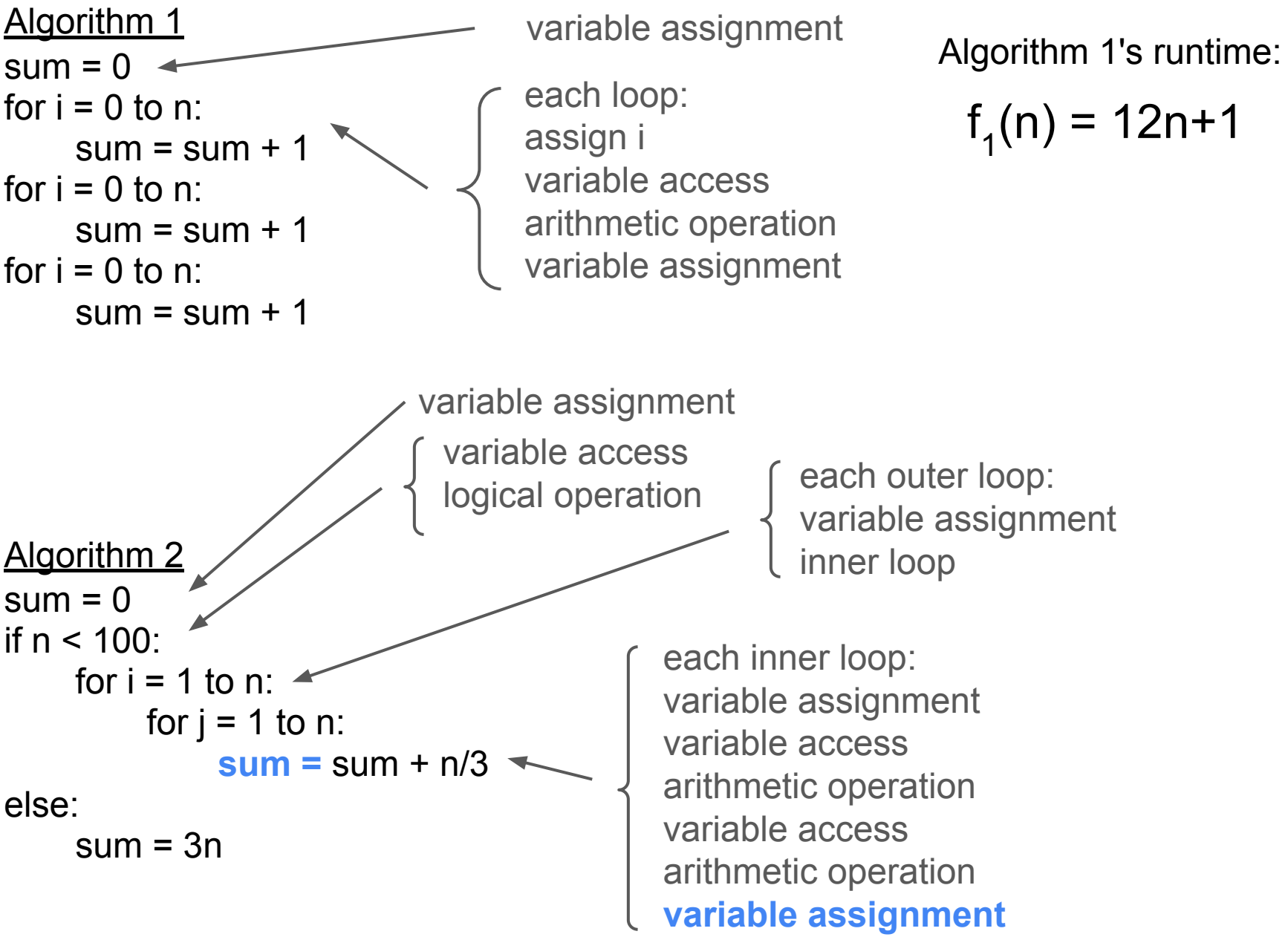
variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$$f_1(n) = 12n+1$$

variable assignment

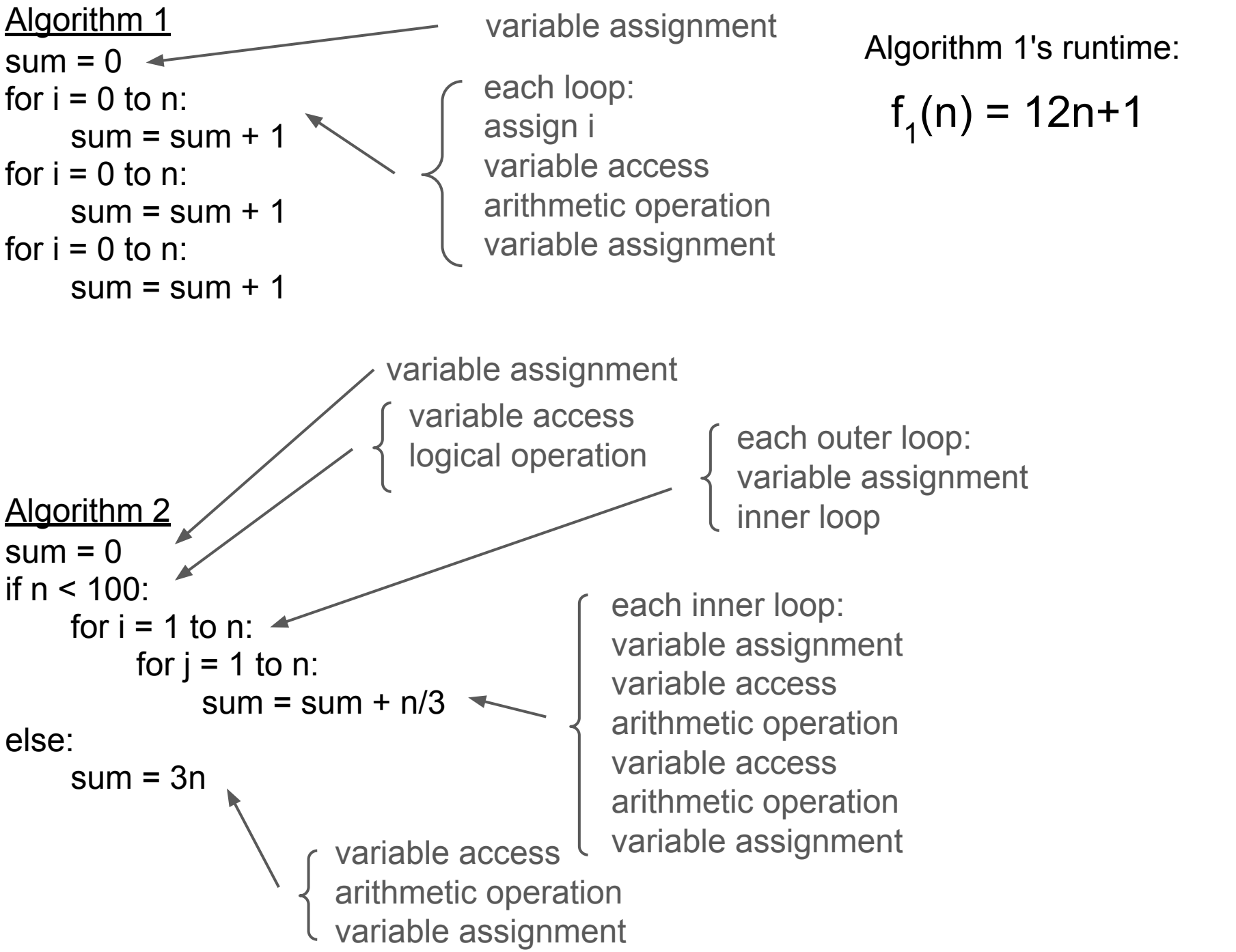
variable access
logical operation

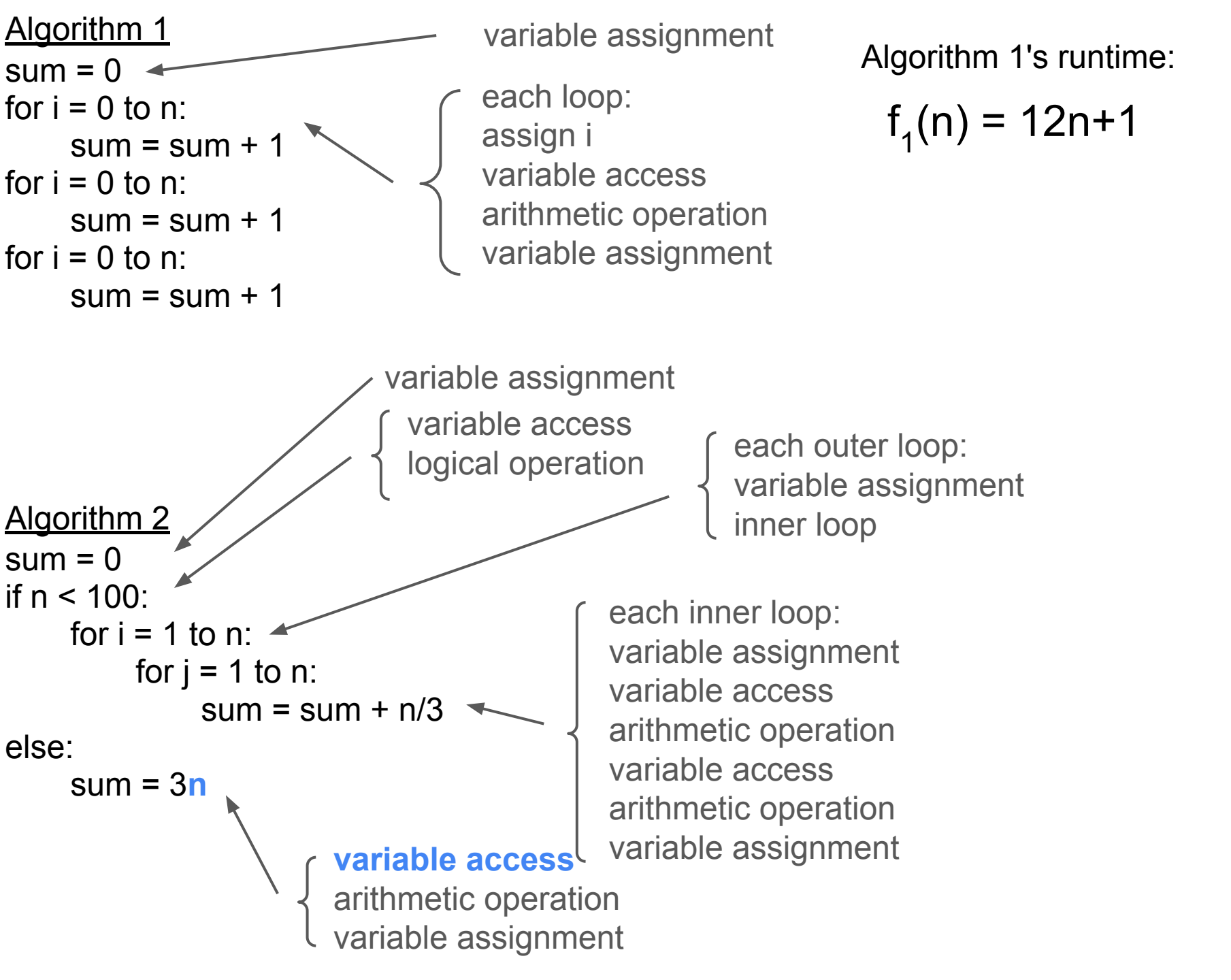
each outer loop:
variable assignment
inner loop

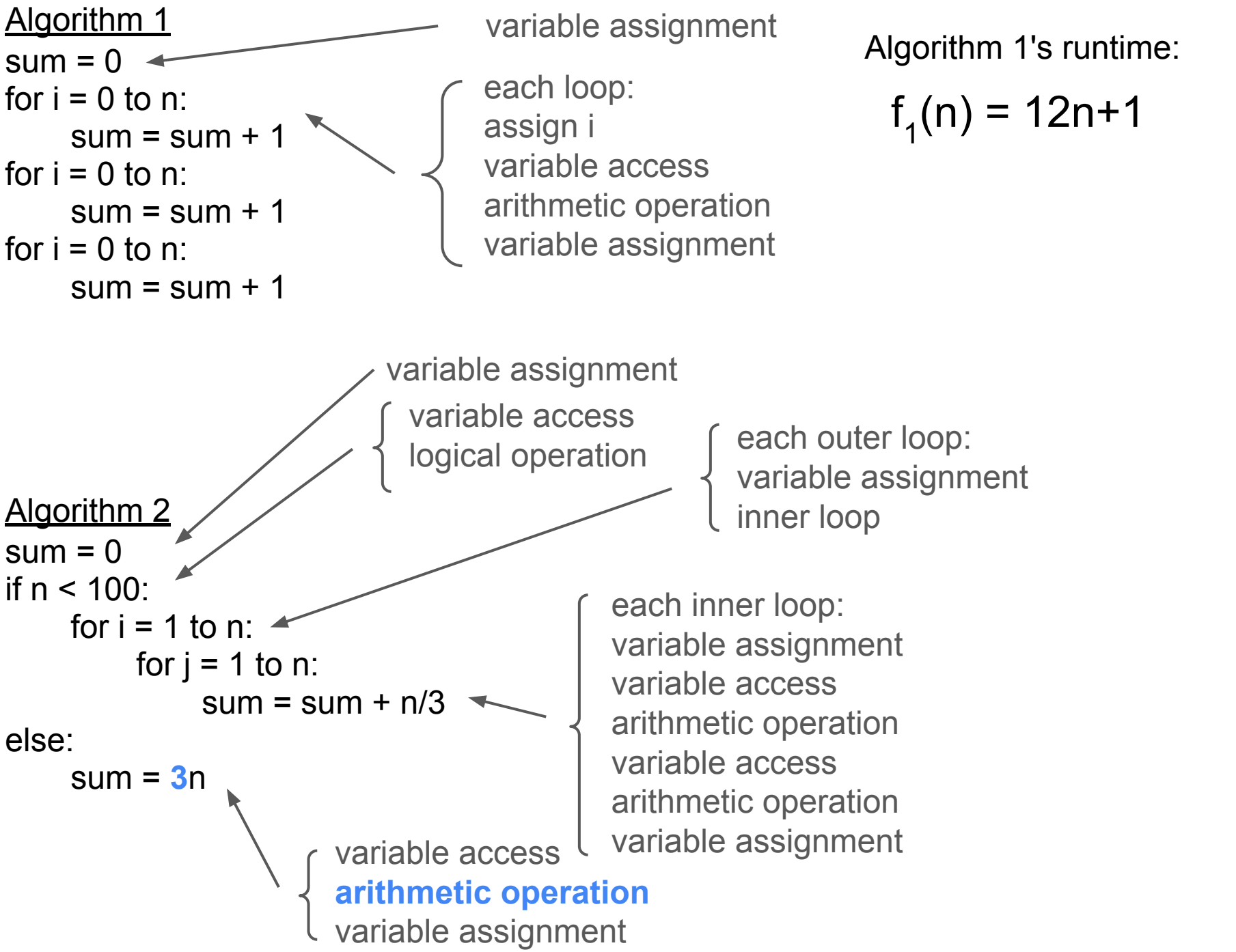
Algorithm 2

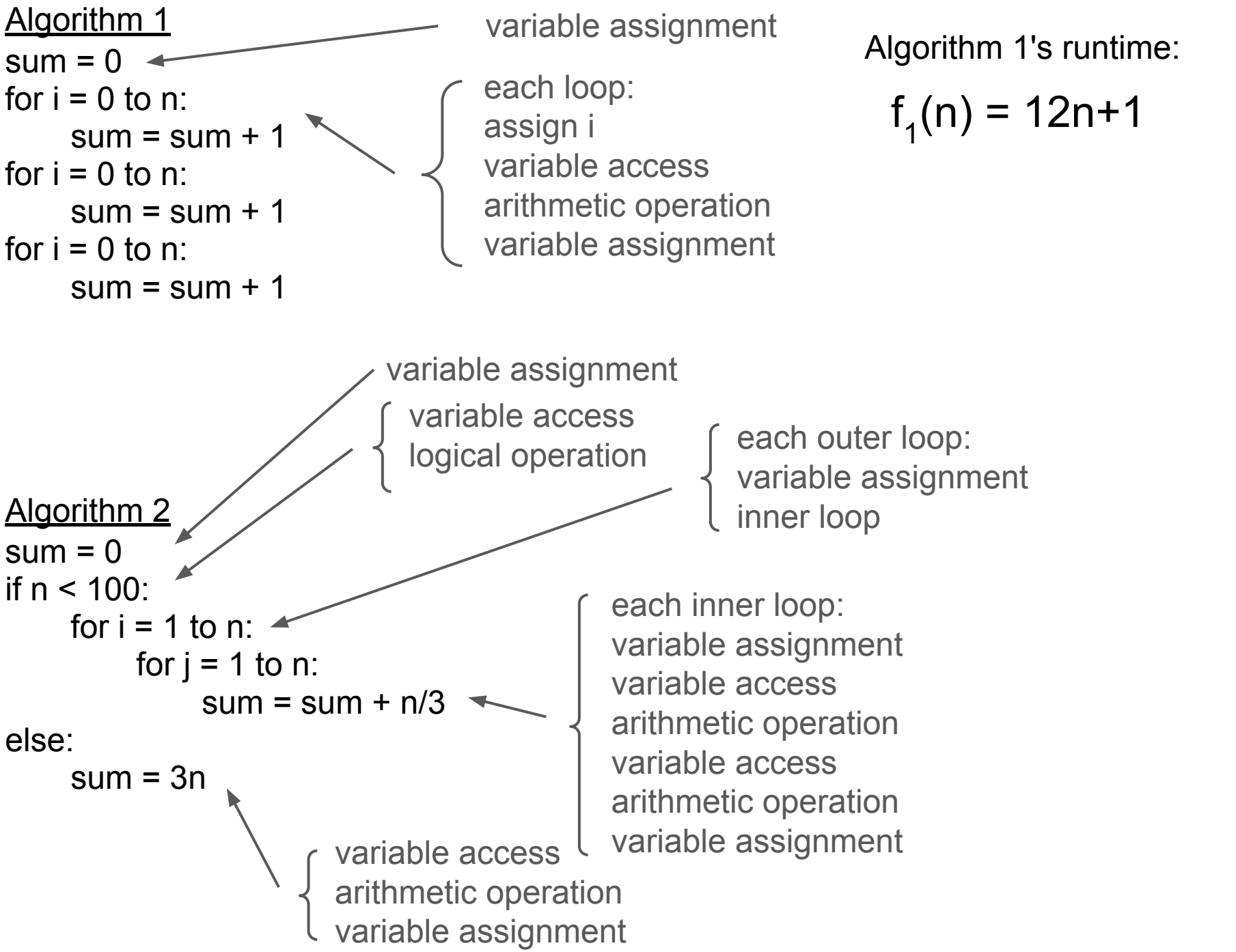
```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

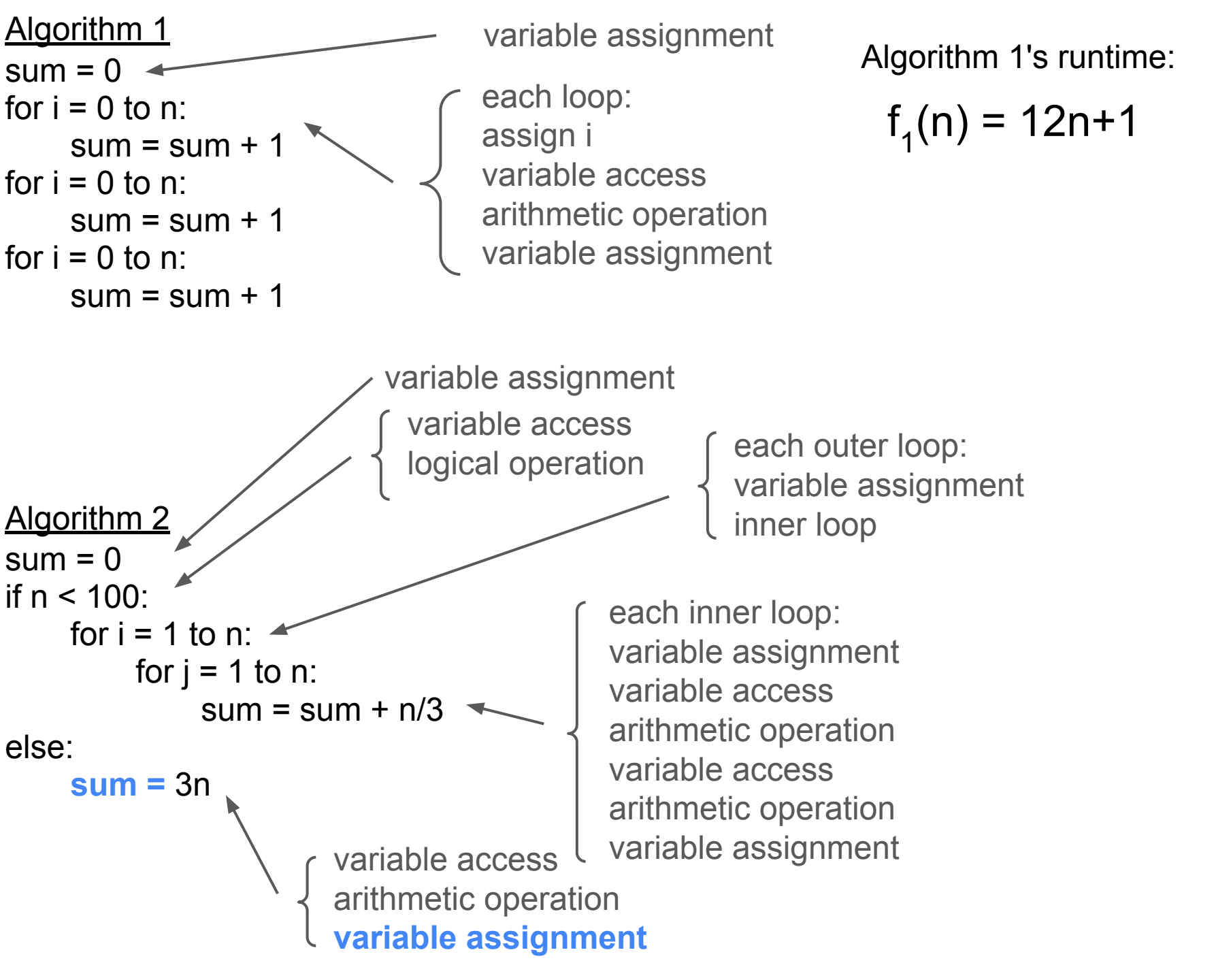
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

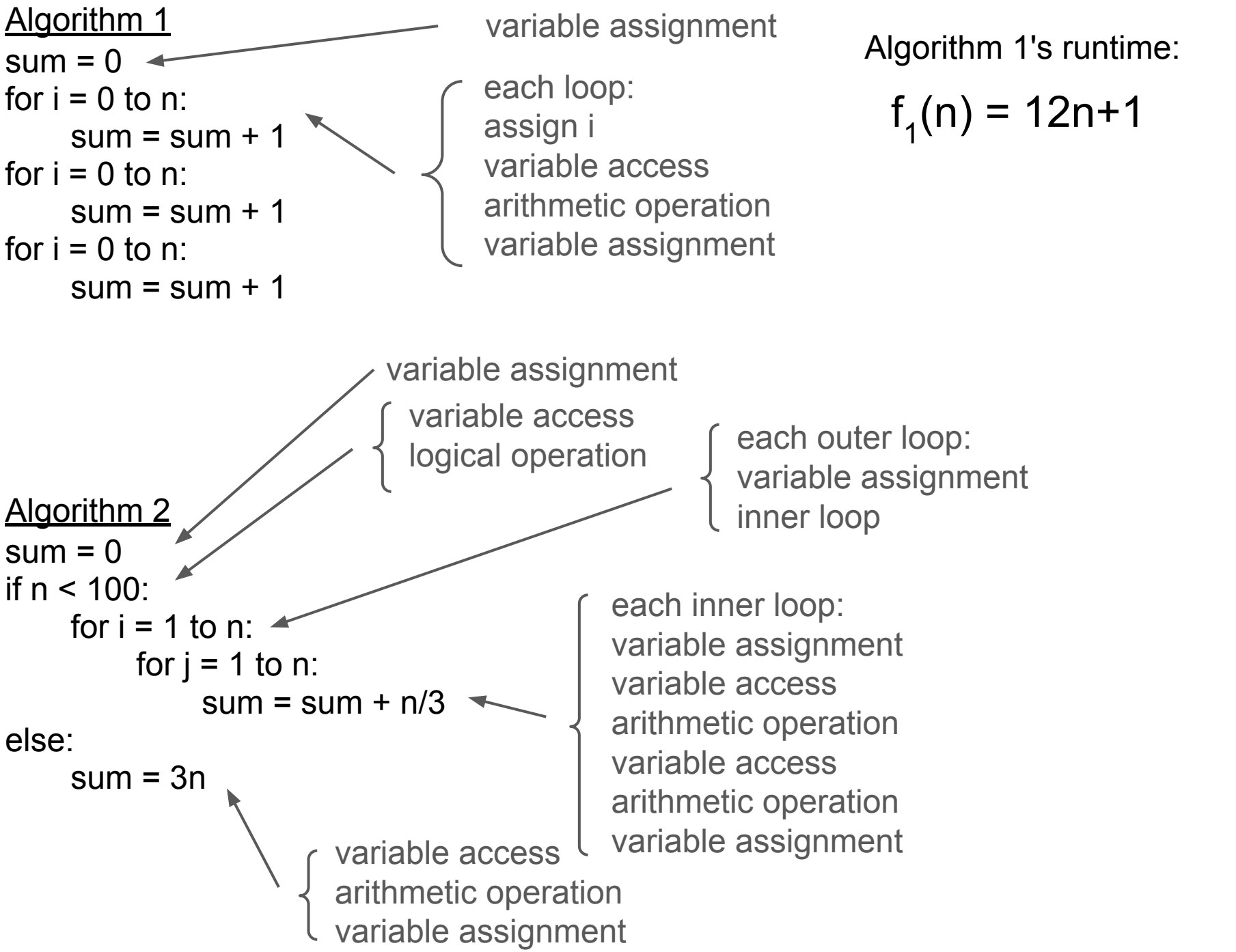

Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment

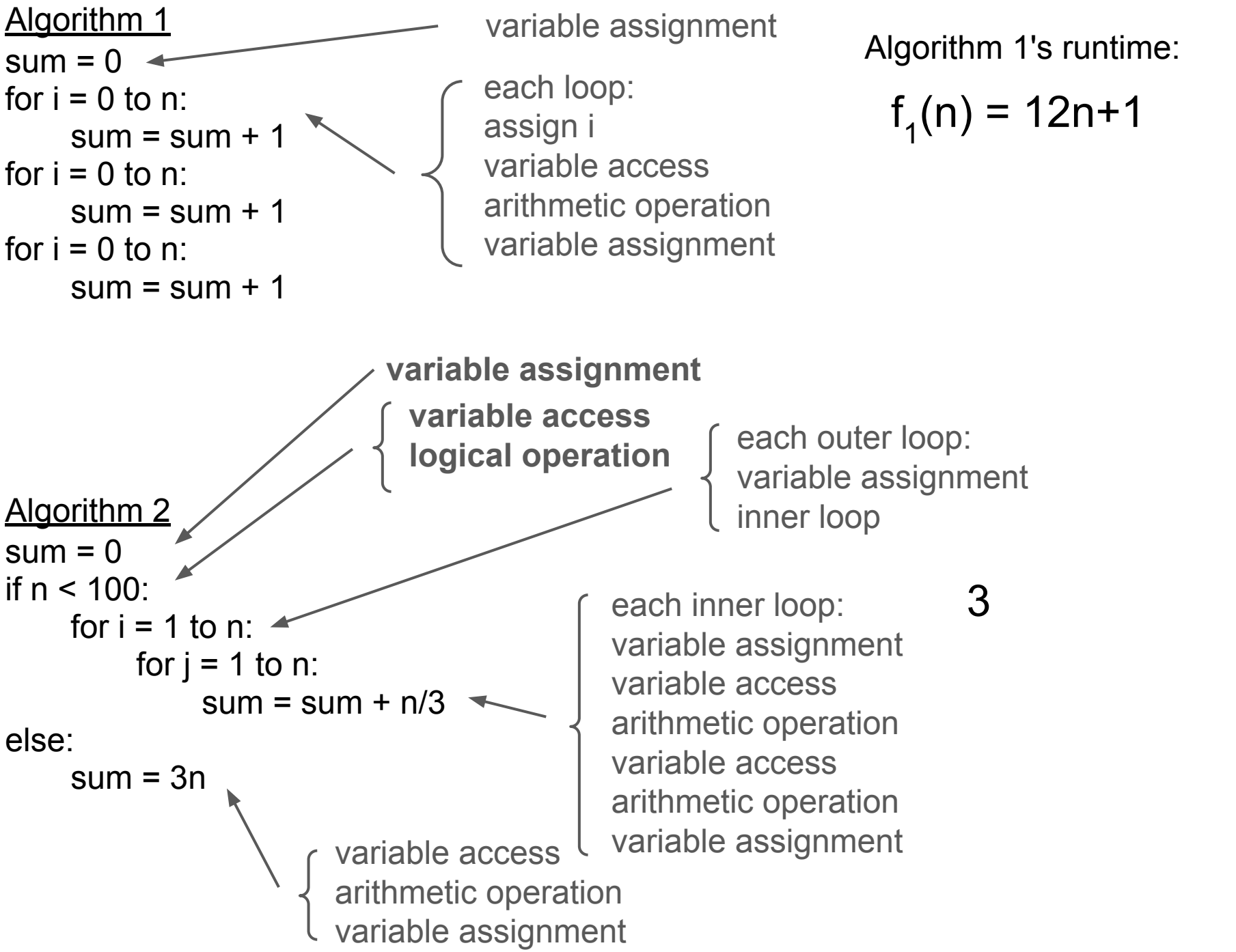

Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment
3

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$f_1(n) = 12n+1$

variable assignment

variable access
logical operation

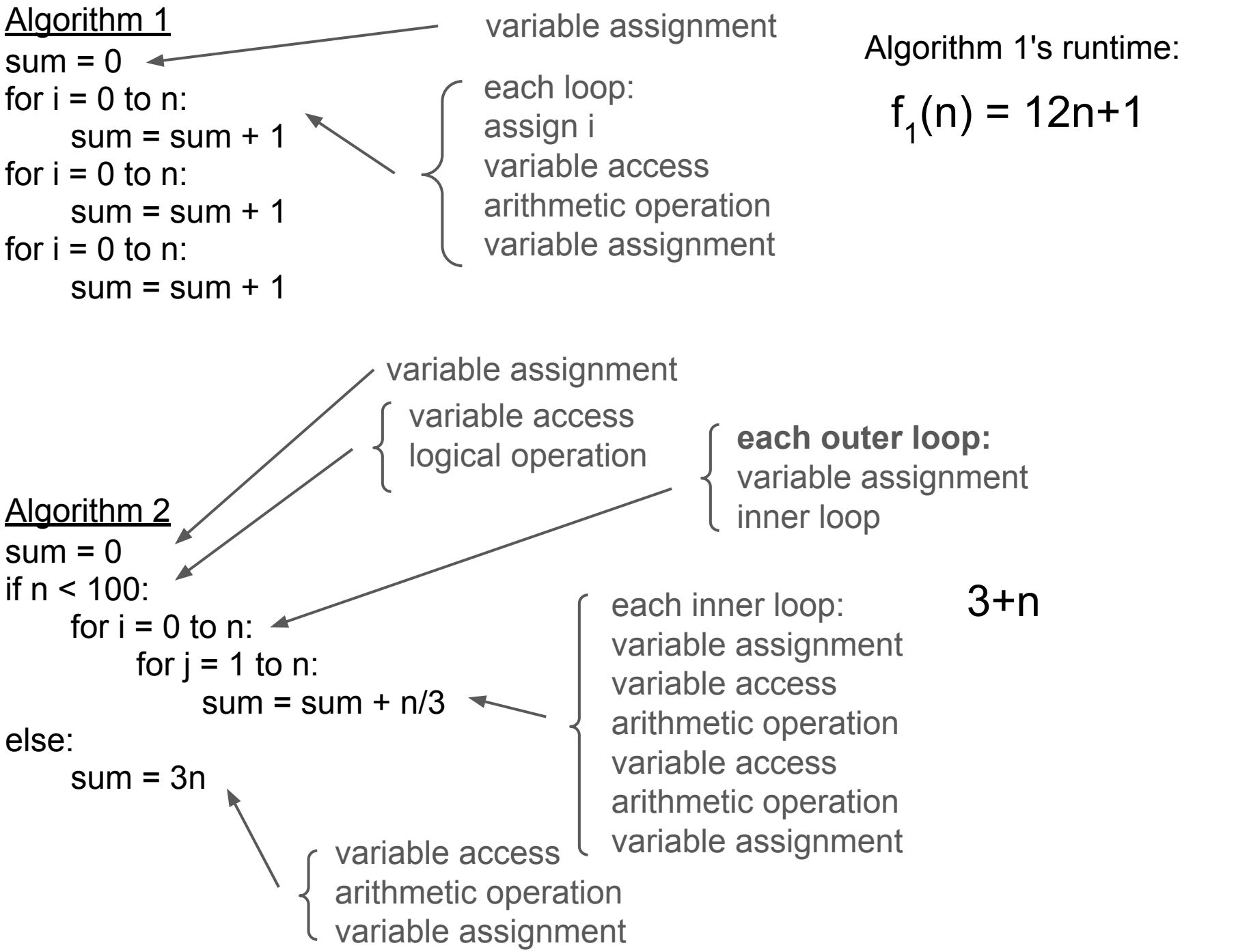
each outer loop:
variable assignment
inner loop

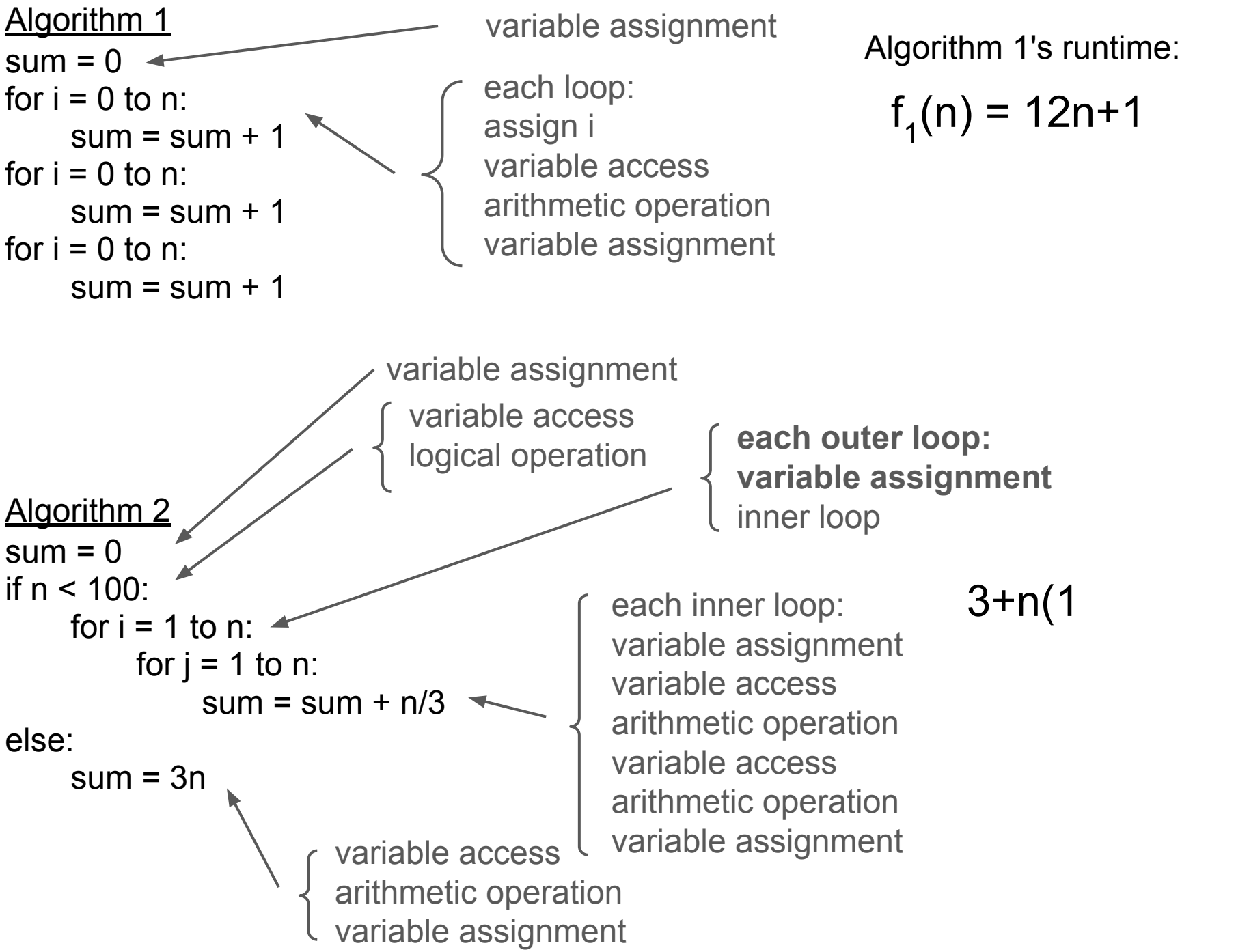
Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 0 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
3+n
variable access
arithmetic operation
variable assignment


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
3+n(1
variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$f_1(n) = 12n+1$

variable assignment

variable access
logical operation

each outer loop:
variable assignment
inner loop

Algorithm 2

```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

$3+n(1+6n)$

variable access
arithmetic operation
variable assignment

Algorithm 1

```
sum = 0
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
for i = 0 to n:
      sum = sum + 1
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

Algorithm 1's runtime:

$f_1(n) = 12n+1$

variable assignment

variable access
logical operation

each outer loop:
variable assignment
inner loop

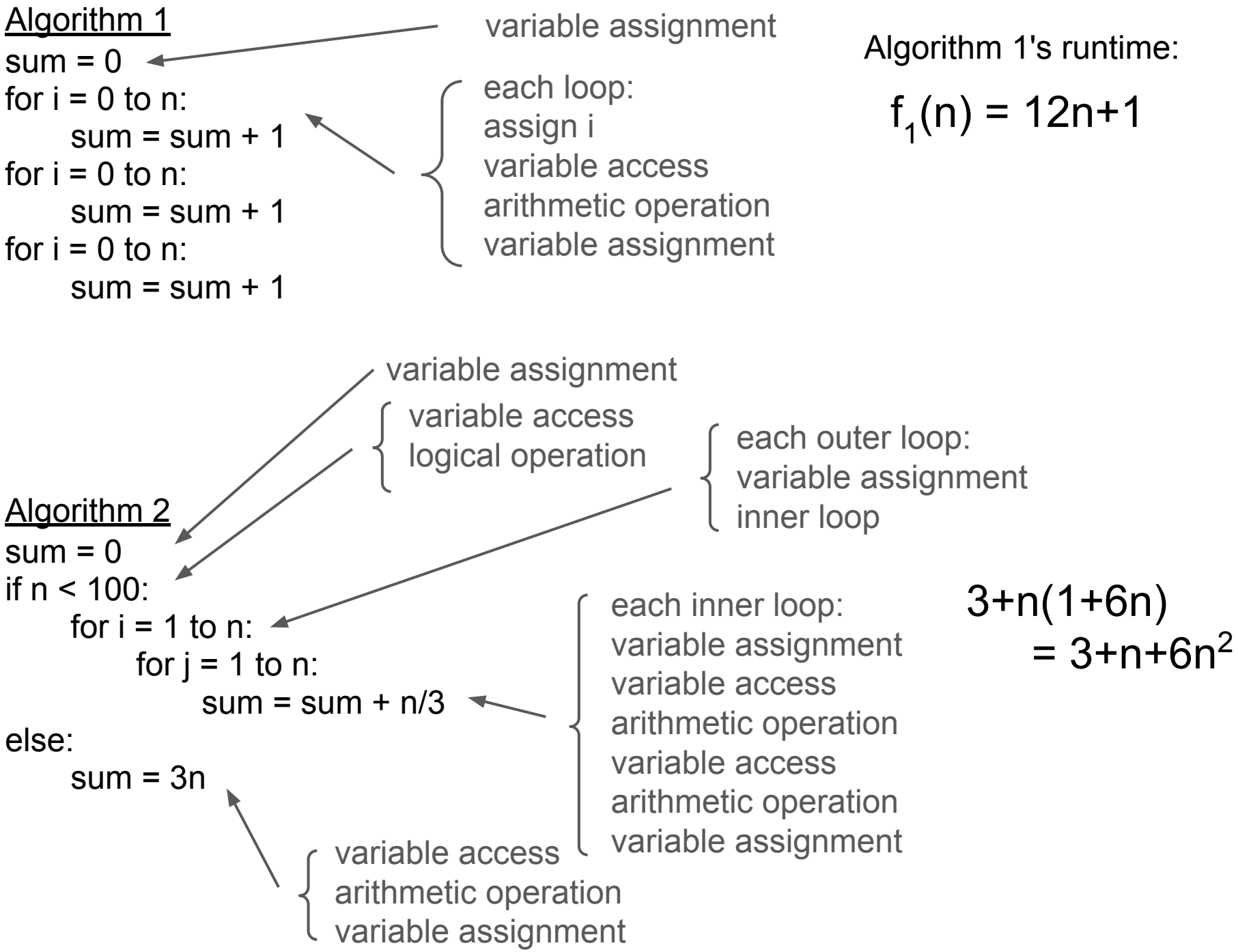
Algorithm 2

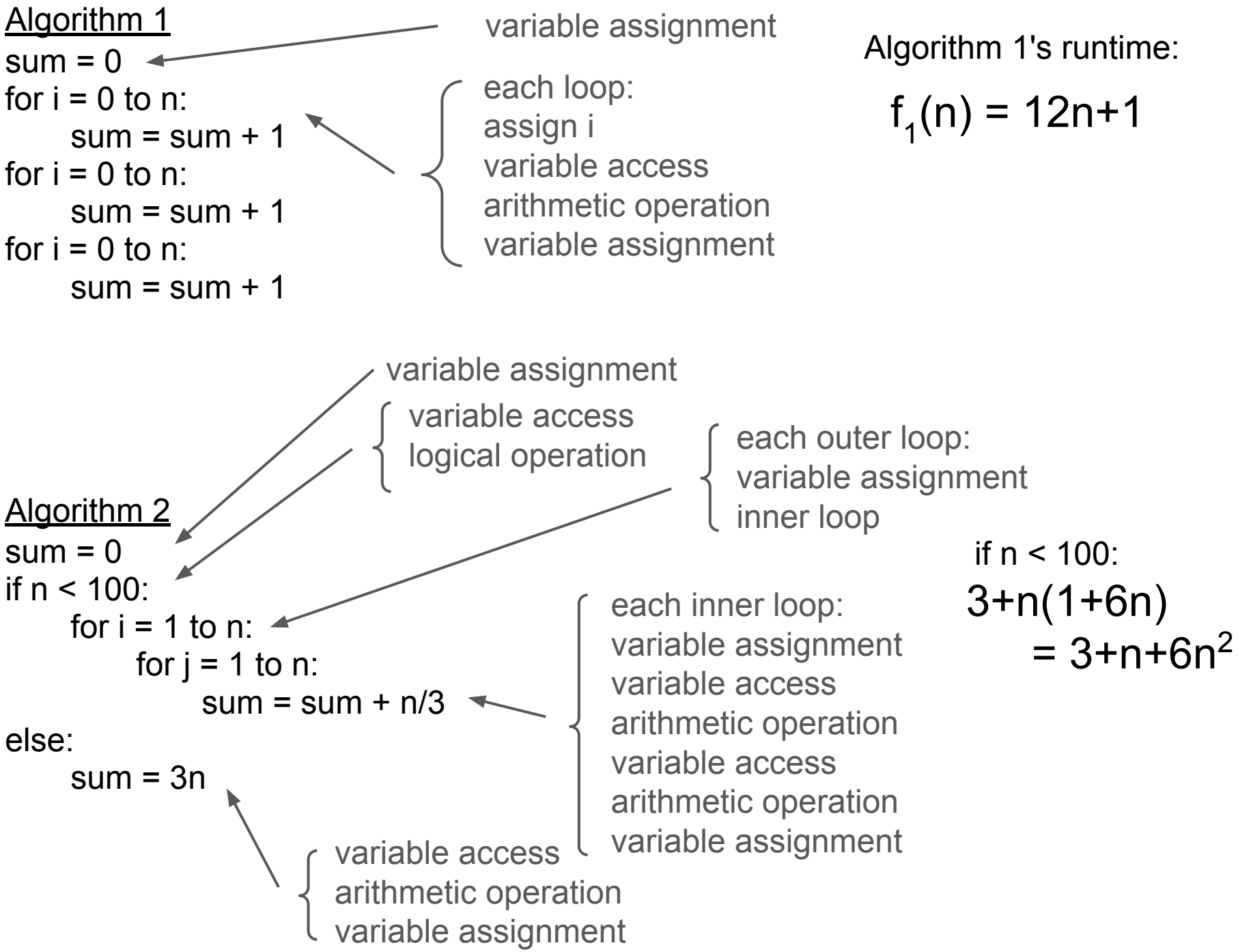
```
sum = 0
if n < 100:
      for i = 1 to n:
            for j = 1 to n:
                  sum = sum + n/3
else:
      sum = 3n
```

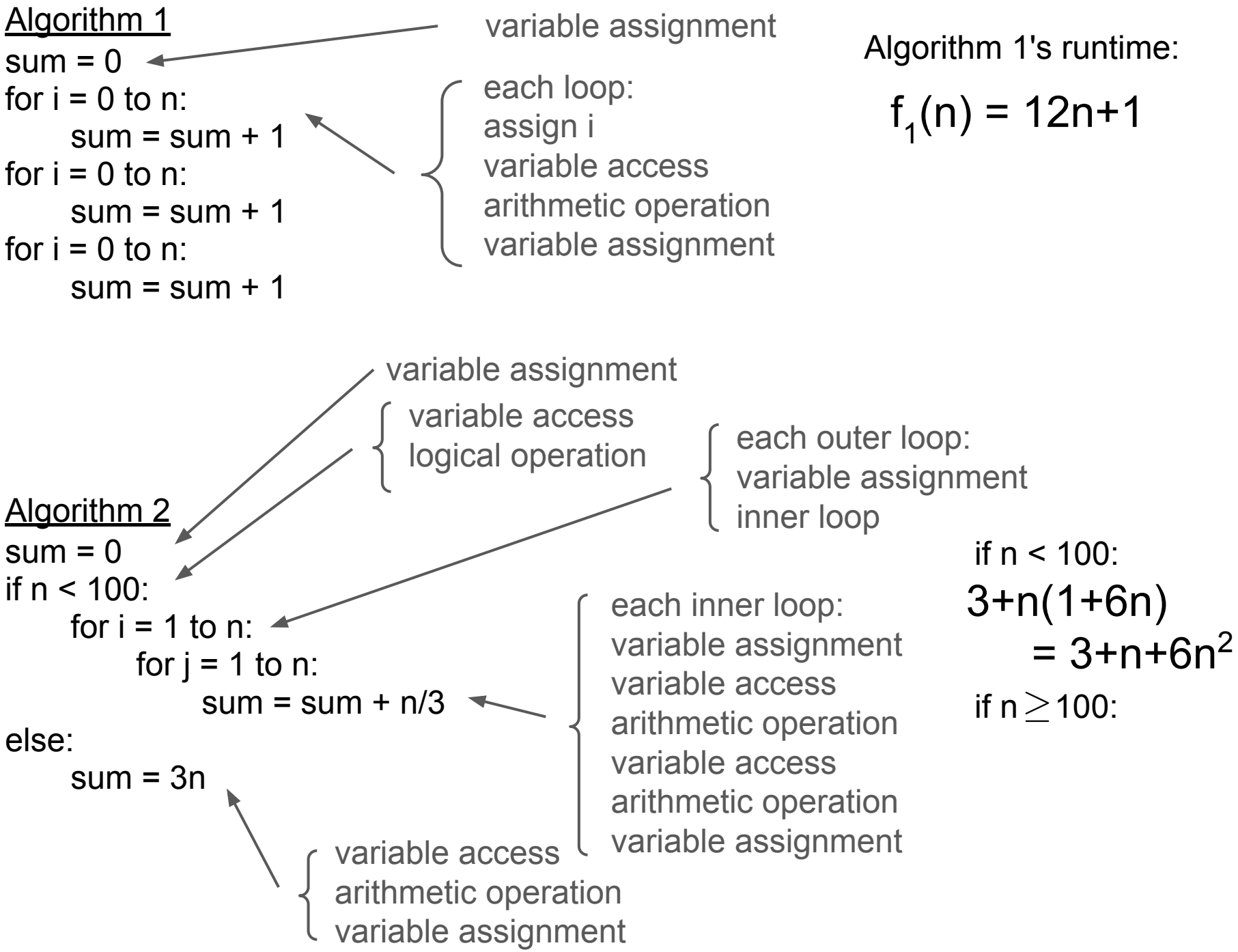
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment

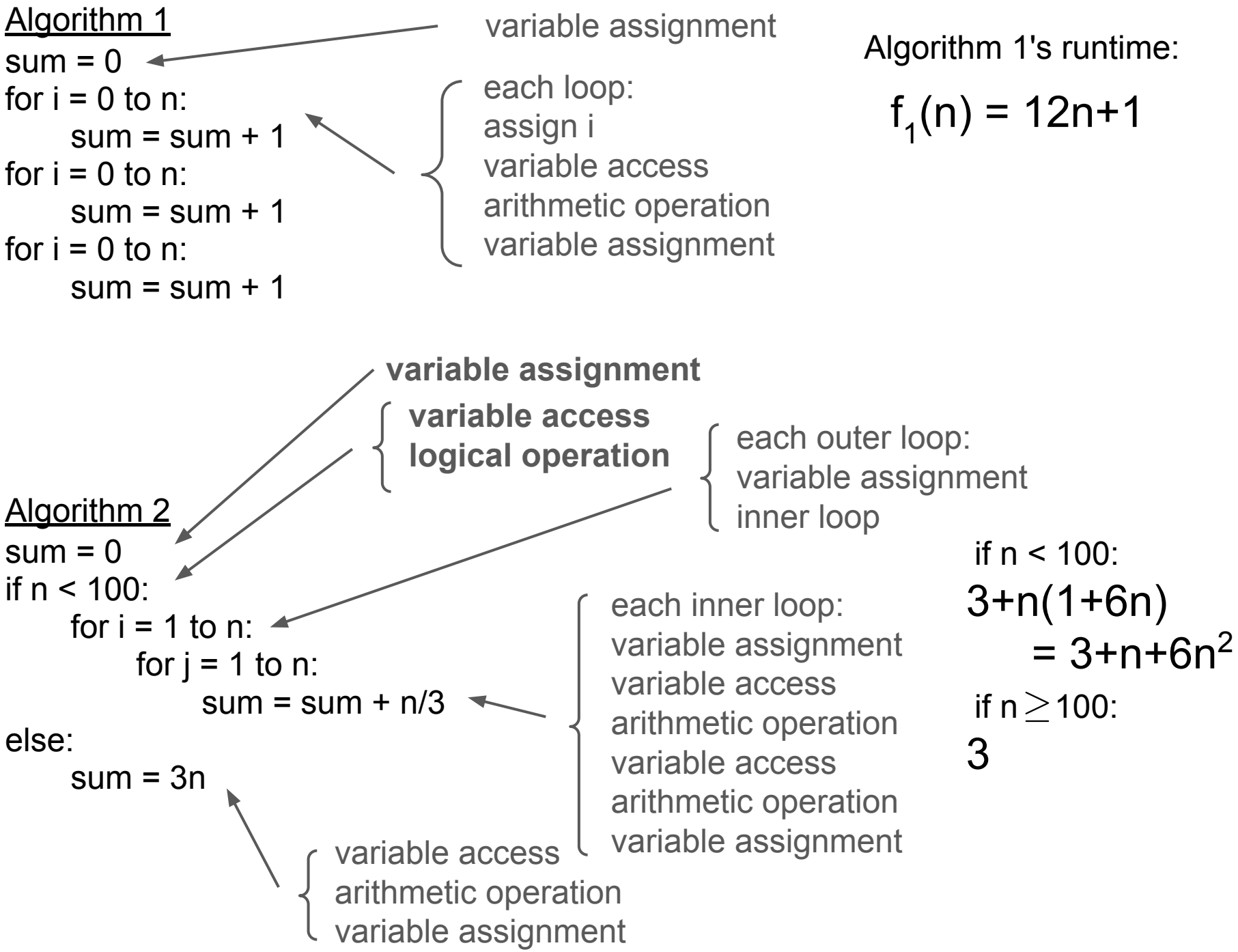
$3+n(1+6n)$

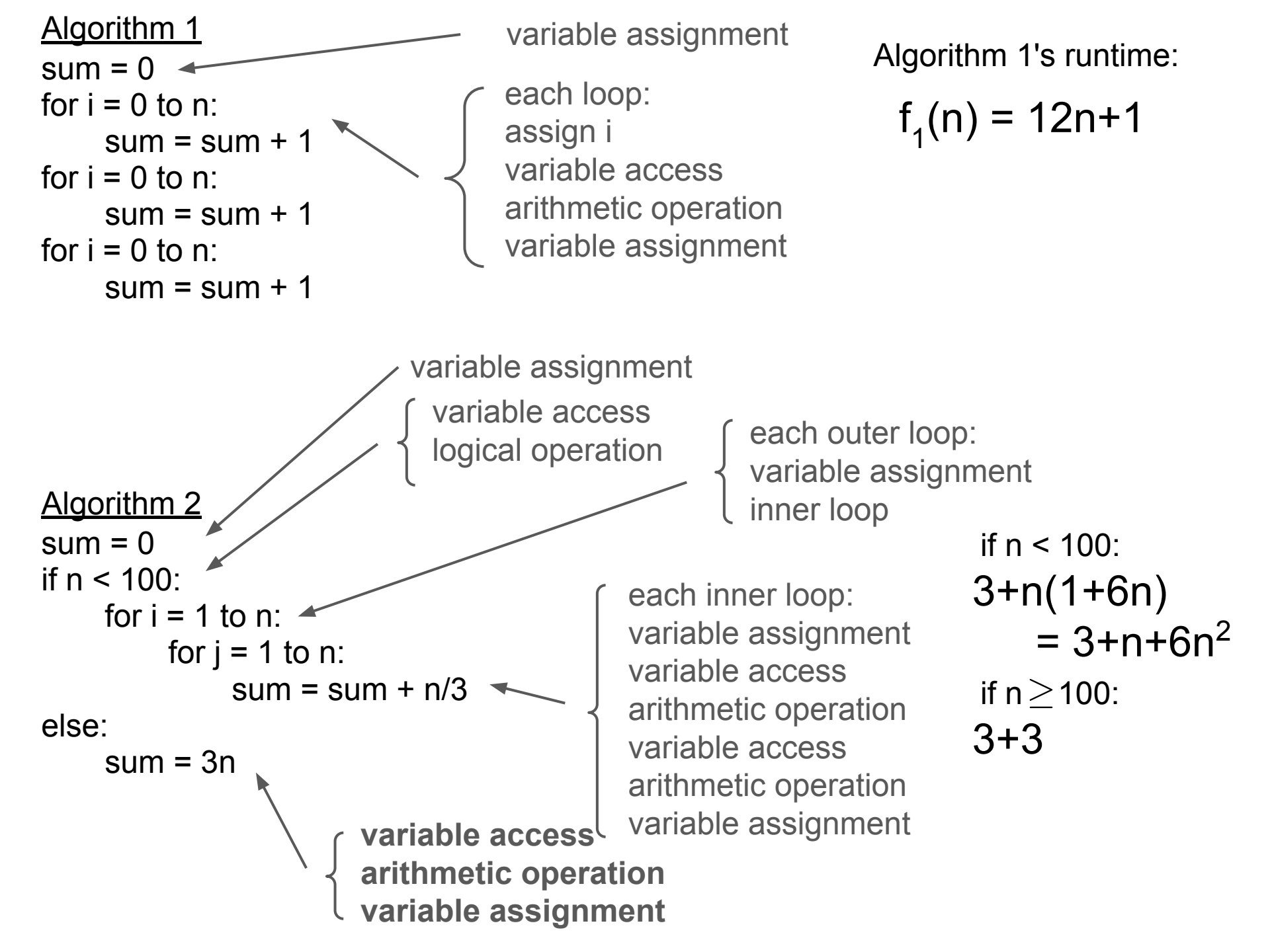
variable access
arithmetic operation
variable assignment

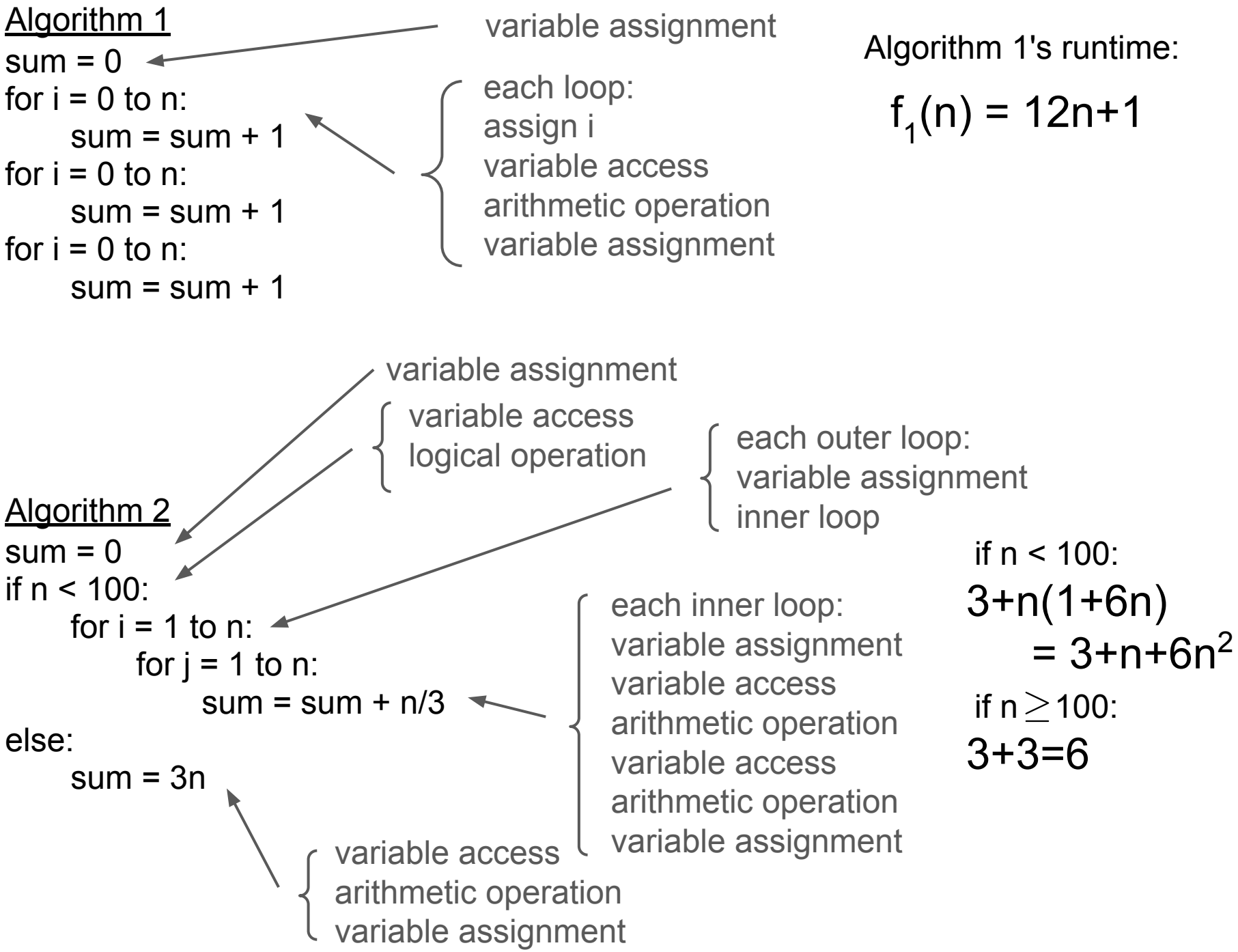

Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
$3+n(1+6n)$
$= 3+n+6n^2$
variable access
arithmetic operation
variable assignment

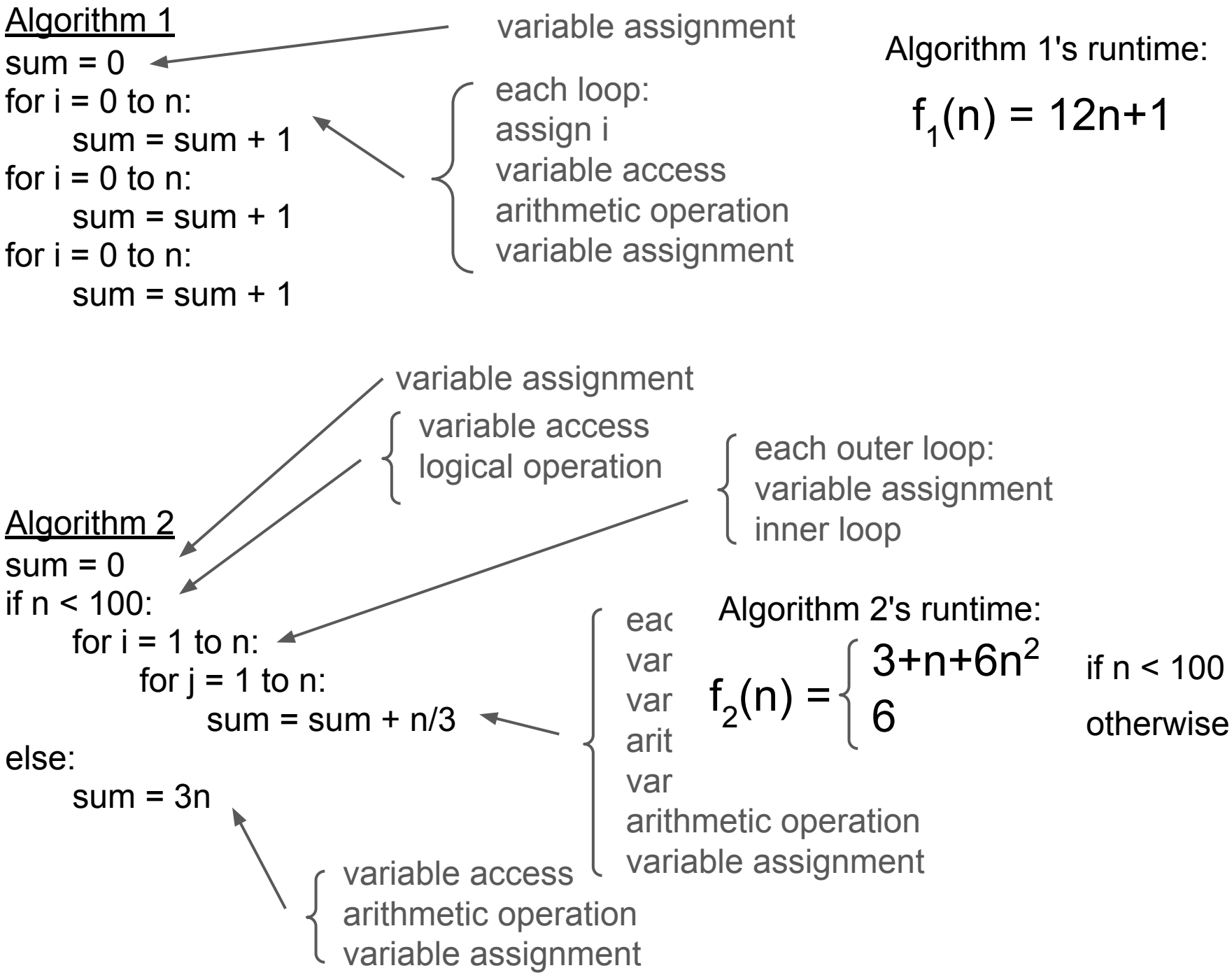

Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment
if n < 100:
$3+n(1+6n)$
$= 3+n+6n^2$


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment
if n < 100:
$3+n(1+6n)$
$= 3+n+6n^2$
if n $\geq$ 100:


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment
if n < 100:
$3+n(1+6n)$
$= 3+n+6n^2$
if $n \geq 100$:
3


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment
if n < 100:
$3+n(1+6n)$
$= 3+n+6n^2$
if $n \geq 100$:
3+3

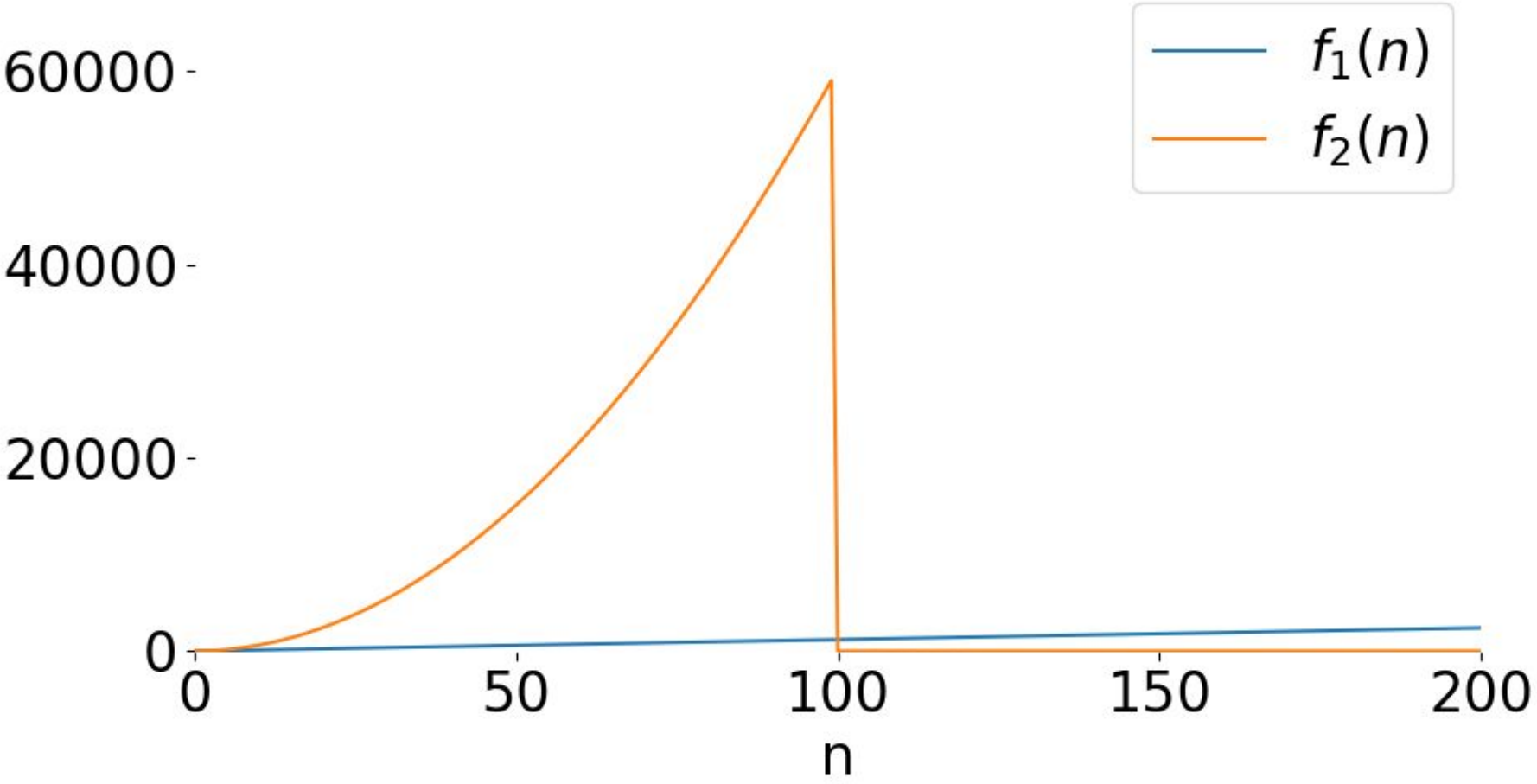

Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
each inner loop:
variable assignment
variable access
arithmetic operation
variable access
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment
if n < 100:
$3+n(1+6n)$
$= 3+n+6n^2$
if $n \geq 100$:
3+3=6


Algorithm 1
sum = 0
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
for i = 0 to n:
sum = sum + 1
variable assignment
each loop:
assign i
variable access
arithmetic operation
variable assignment
Algorithm 1's runtime:
$f_1(n) = 12n+1$
variable assignment
variable access
logical operation
each outer loop:
variable assignment
inner loop
Algorithm 2
sum = 0
if n < 100:
for i = 1 to n:
for j = 1 to n:
sum = sum + n/3
else:
sum = 3n
Algorithm 2's runtime:
$f_2(n) = \begin{cases} 3+n+6n^2 & \text{if } n < 100 \\ 6 & \text{otherwise} \end{cases}$
arithmetic operation
variable assignment
variable access
arithmetic operation
variable assignment

Is Algorithm 1 or Algorithm 2 faster?

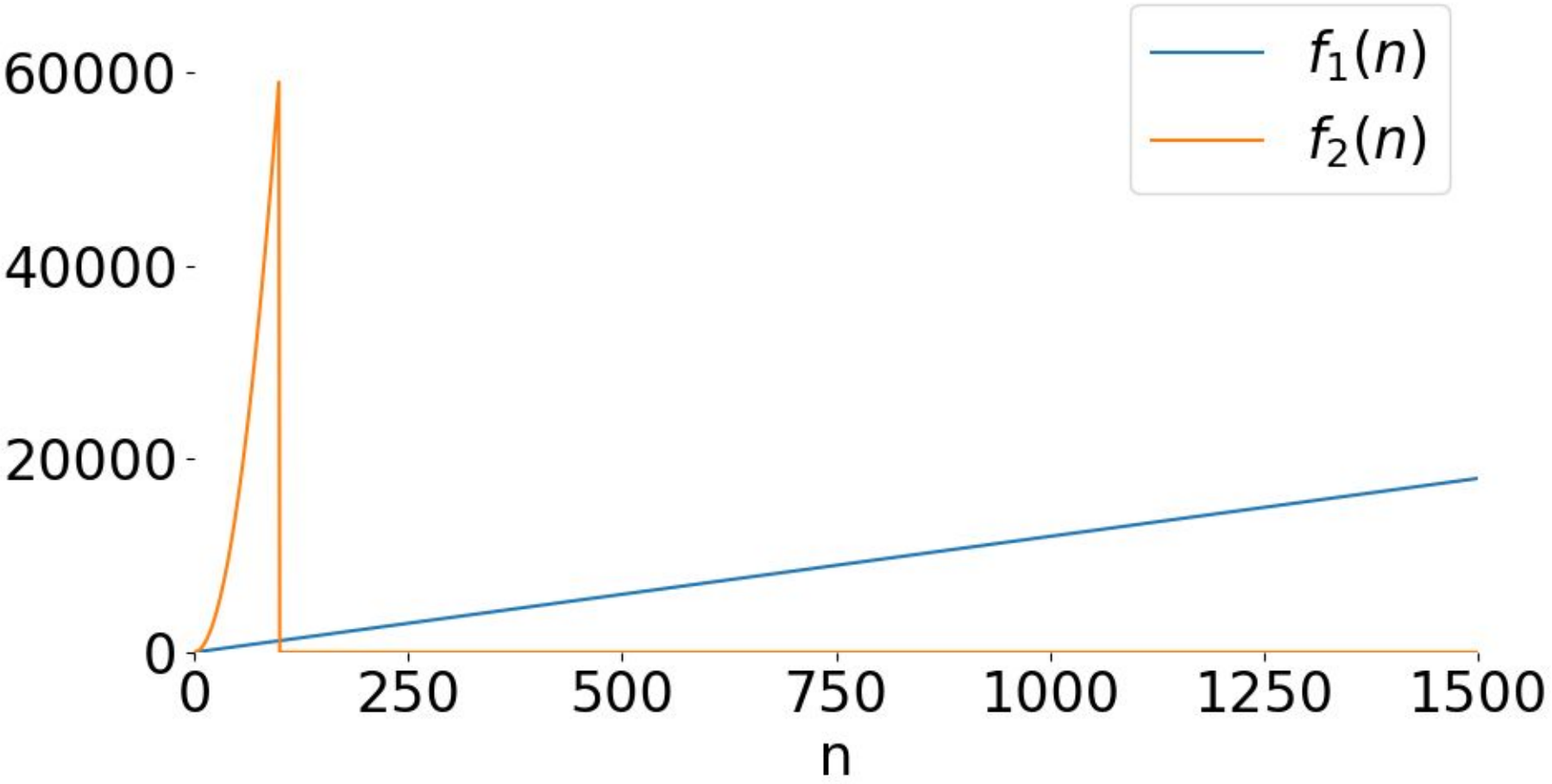
$$f_1(n) = 12n + 1$$

$$f_2(n) = \begin{cases} 6n^2 + 2n + 3 & \text{if } n < 100, \\ 6 & \text{otherwise} \end{cases}$$



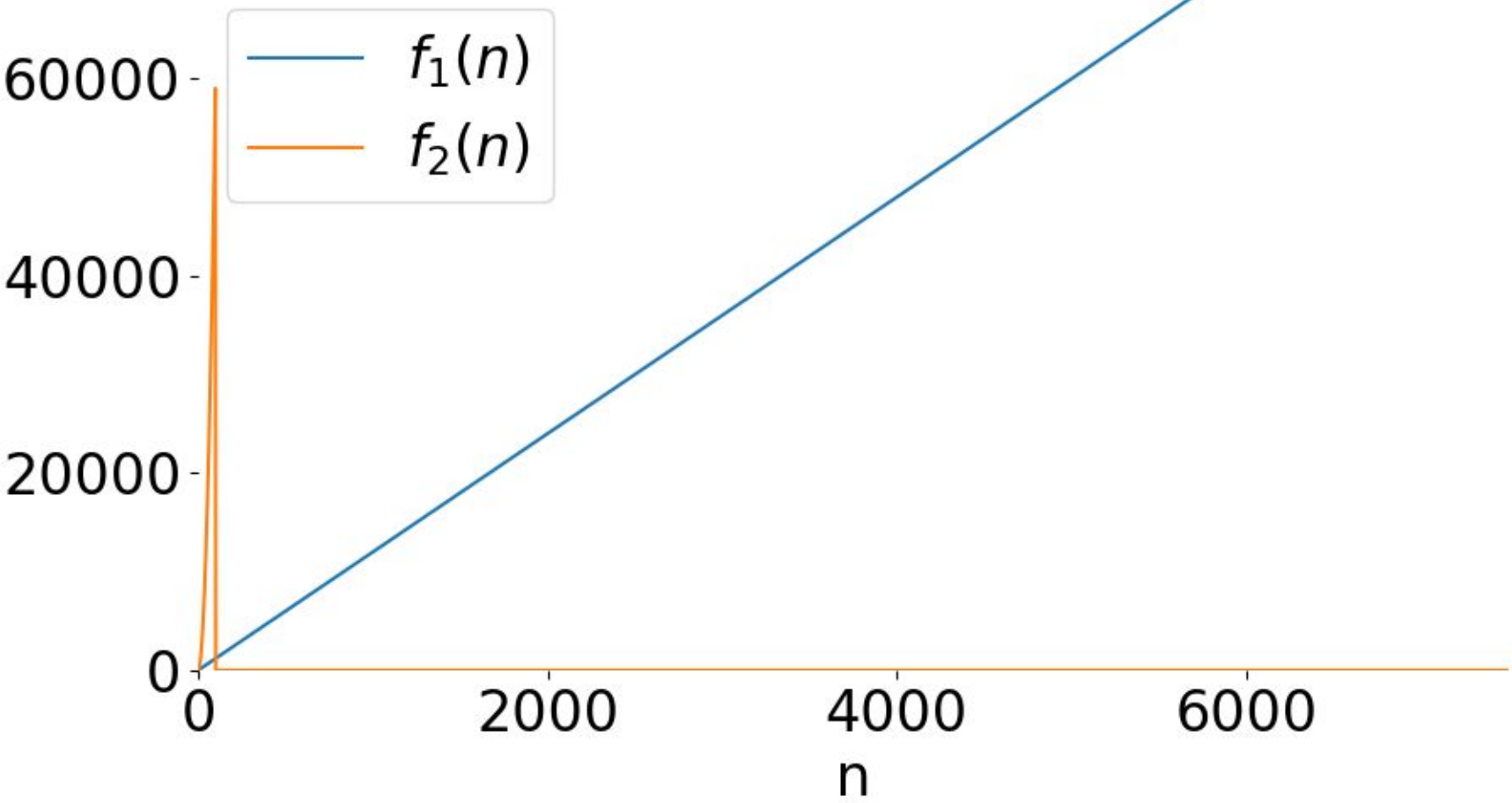
$$f_1(n) = 12n + 1$$

$$f_2(n) = \begin{cases} 6n^2 + 2n + 3 & \text{if } n < 100, \\ 6 & \text{otherwise} \end{cases}$$



$$f_1(n) = 12n + 1$$

$$f_2(n) = \begin{cases} 6n^2 + 2n + 3 & \text{if } n < 100, \\ 6 & \text{otherwise} \end{cases}$$



$$f_1(n) = 12n + 1$$

$$f_2(n) = \begin{cases} 6n^2 + 2n + 3 & \text{if } n < 100, \\ 6 & \text{otherwise} \end{cases}$$

<u>Algorithm 3</u>

```
sum = 0
for i = 1 to n:
        sum = sum + 3
```

variable assignment

each loop:
assign i
variable access
arithmetic operation
variable assignment

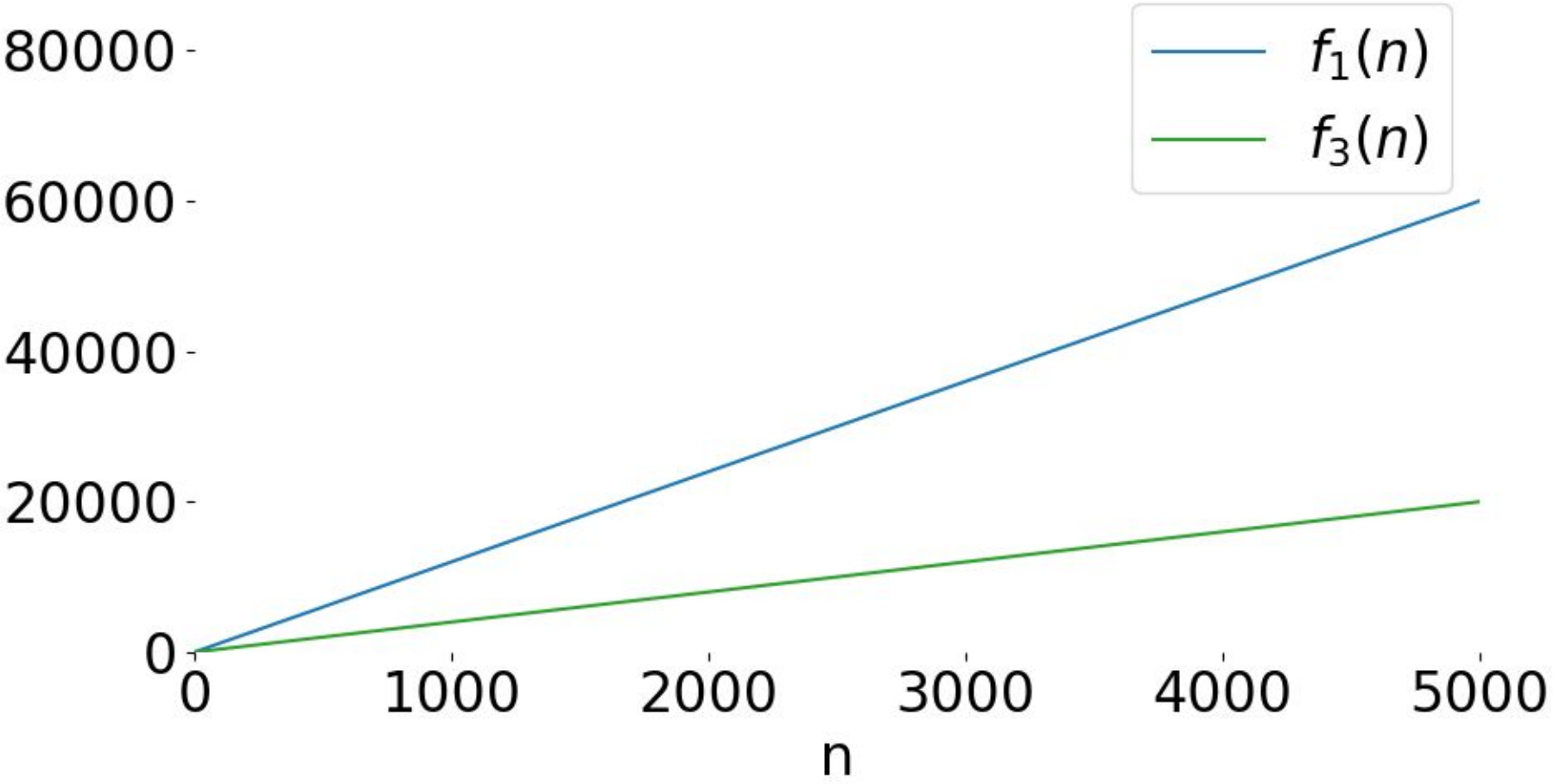
Algorithm 3's runtime:

$f_3(n) = 4n+1$

Is Algorithm 1 or Algorithm 3 faster?

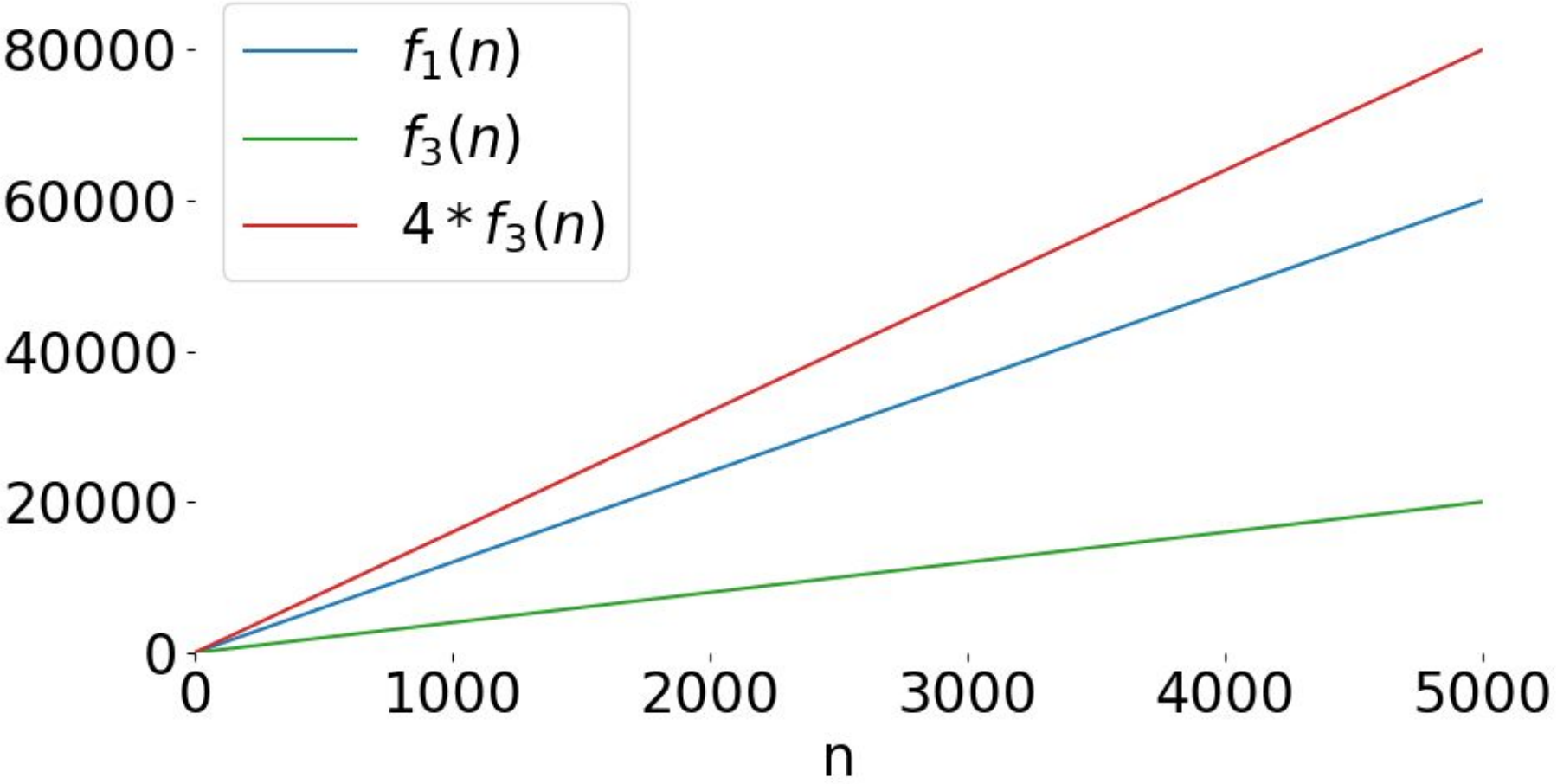
$$f_1(n) = 12n + 1$$

$$f_3(n) = 4n + 1$$



$$f_1(n) = 12n + 1$$

$$f_3(n) = 4n + 1$$



$$f_1(n) = 12n + 1$$

$$f_3(n) = 4n + 1$$