

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

Slides by **Sean Szumlanski**  
for **CS106B**, Programming Abstractions

*Winter 2024*

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

Initially, consider the first element to constitute a separate, **sorted** vector.

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

Initially, consider the first element to constitute a separate, **sorted** vector.

# Insertion Sort

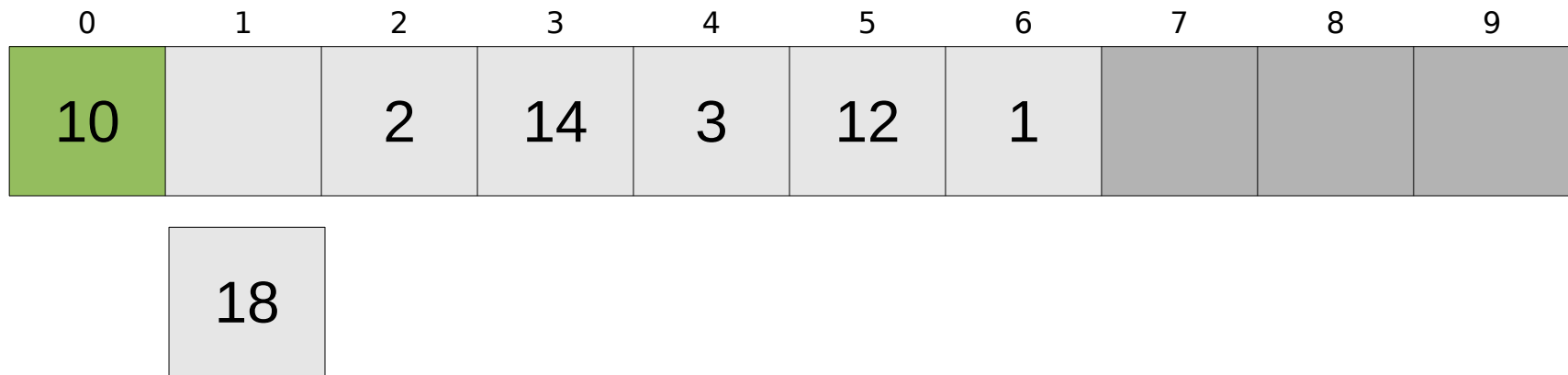
(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

Now pull the first element out of the **unsorted** partition of the vector.

# Insertion Sort

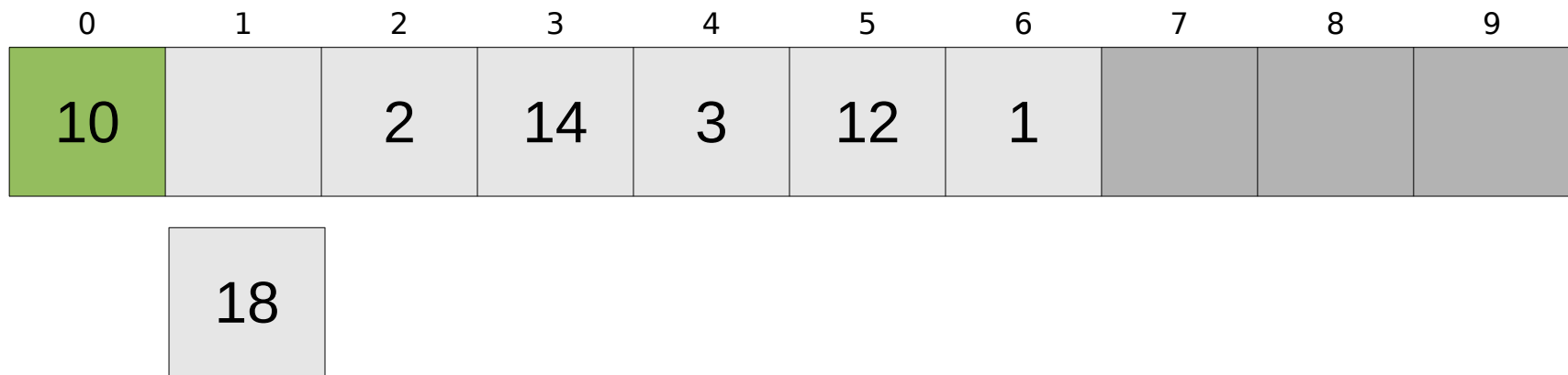
(insert each element into a sorted partition)



Now pull the first element out of the **unsorted** partition of the vector.

# Insertion Sort

(insert each element into a sorted partition)

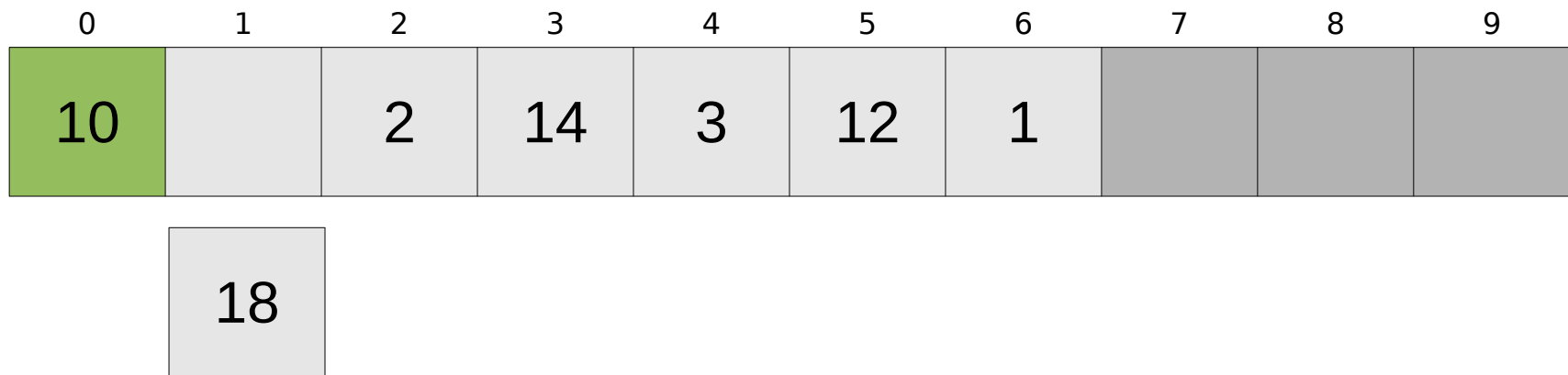


Now pull the first element out of the **unsorted** partition of the vector.

In the **sorted** partition, scooch everything over that is **greater** than that element.

# Insertion Sort

(insert each element into a sorted partition)



Now pull the first element out of the **unsorted** partition of the vector.

In the **sorted** partition, scooch everything over that is **greater** than that element.

Stick the element into the hole left for it in the **sorted** portion of the vector.



# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

Now pull the first element out of the **unsorted** partition of the vector.

In the **sorted** partition, scooch everything over that is **greater** than that element.

Stick the element into the hole left for it in the **sorted** portion of the vector.

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

**That was easy!**

Let's see if we can do it again.

# Insertion Sort

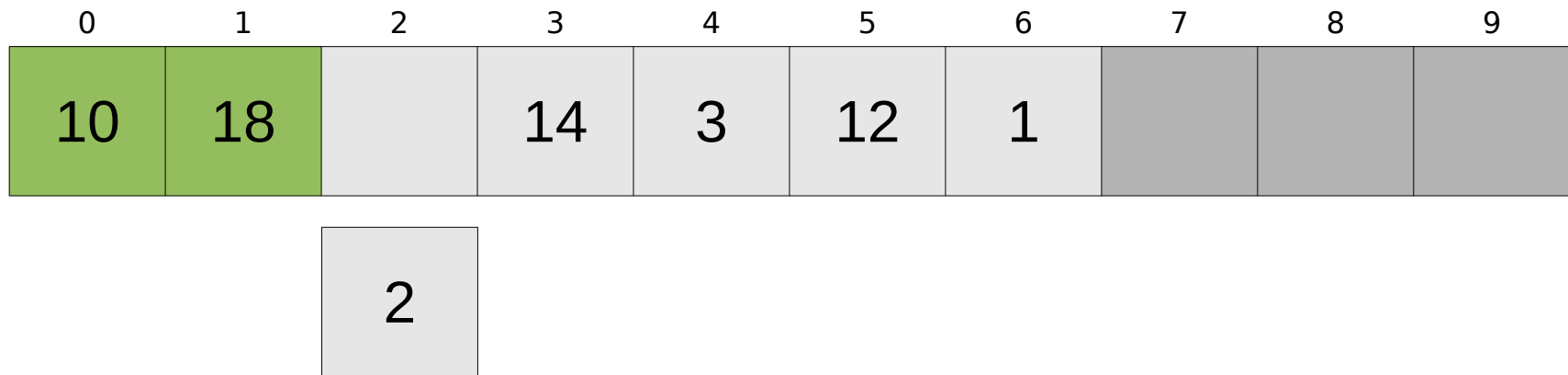
(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
10	18	2	14	3	12	1			

Pull the first element out of the **unsorted** partition of the vector.

# Insertion Sort

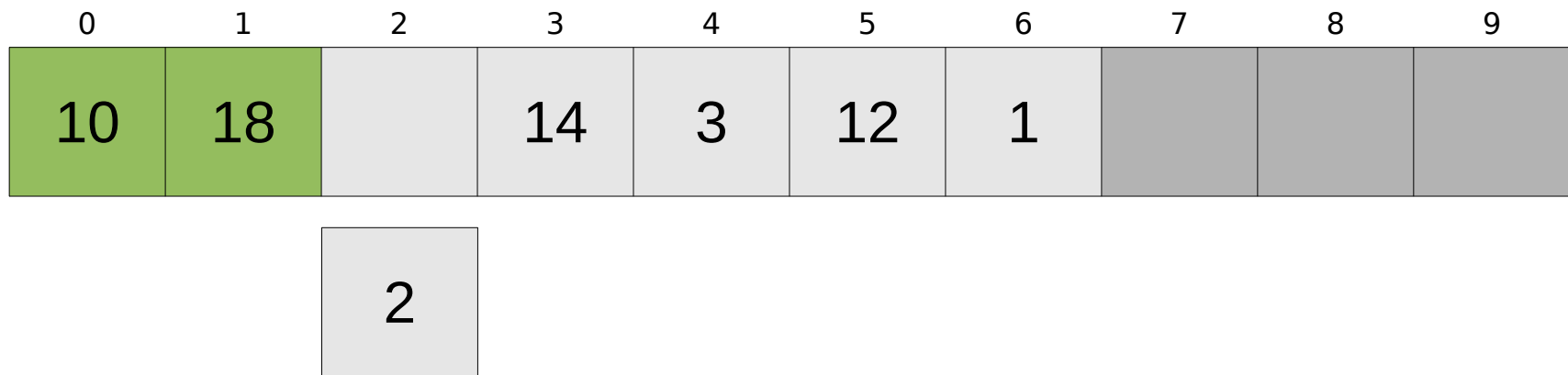
(insert each element into a sorted partition)



Pull the first element out of the **unsorted** partition of the vector.

# Insertion Sort

(insert each element into a sorted partition)

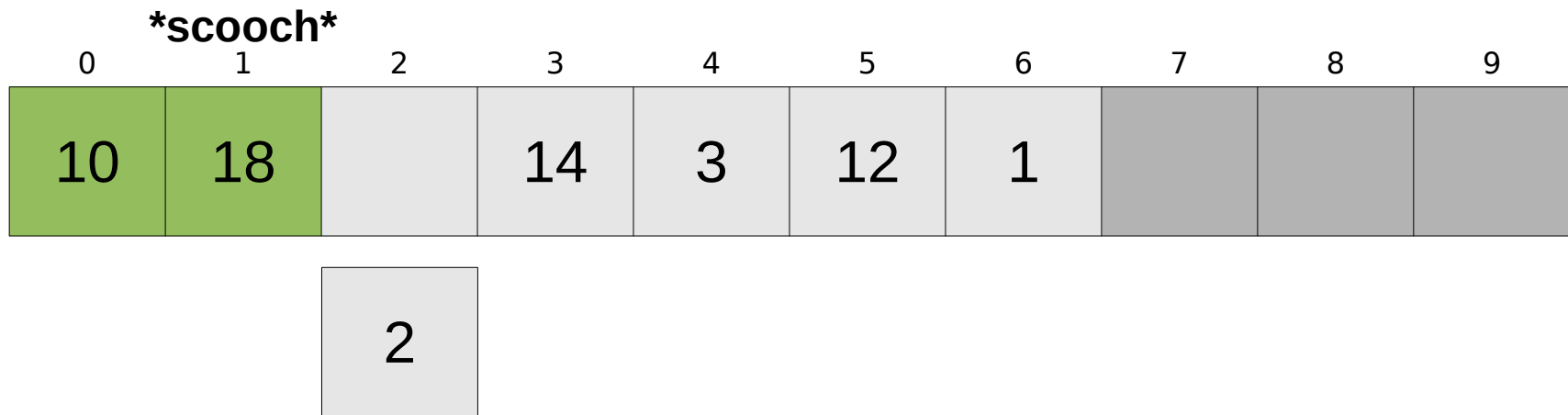


Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

# Insertion Sort

(insert each element into a sorted partition)

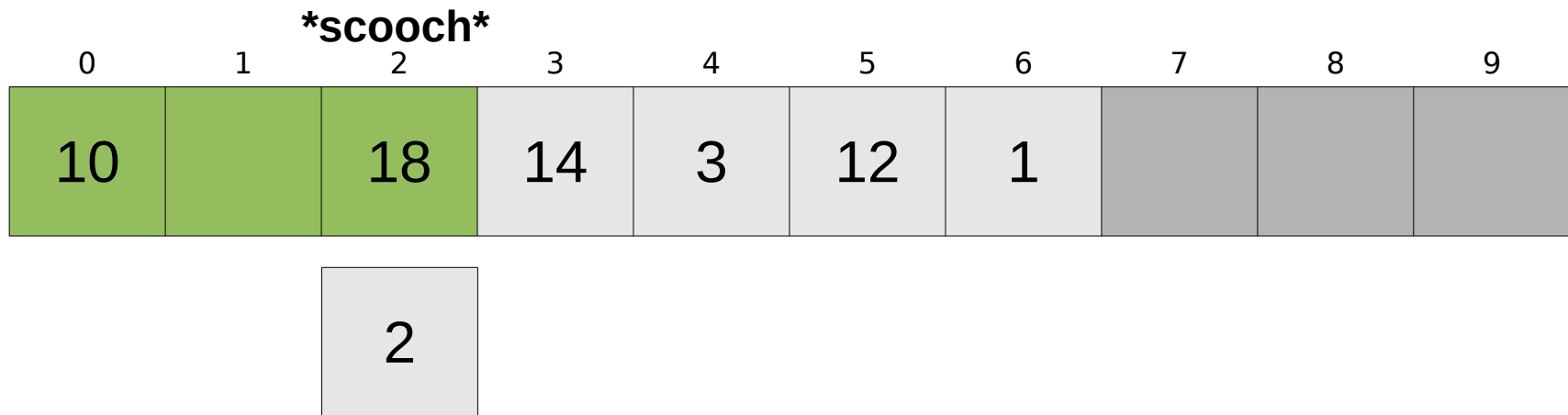


Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

# Insertion Sort

(insert each element into a sorted partition)

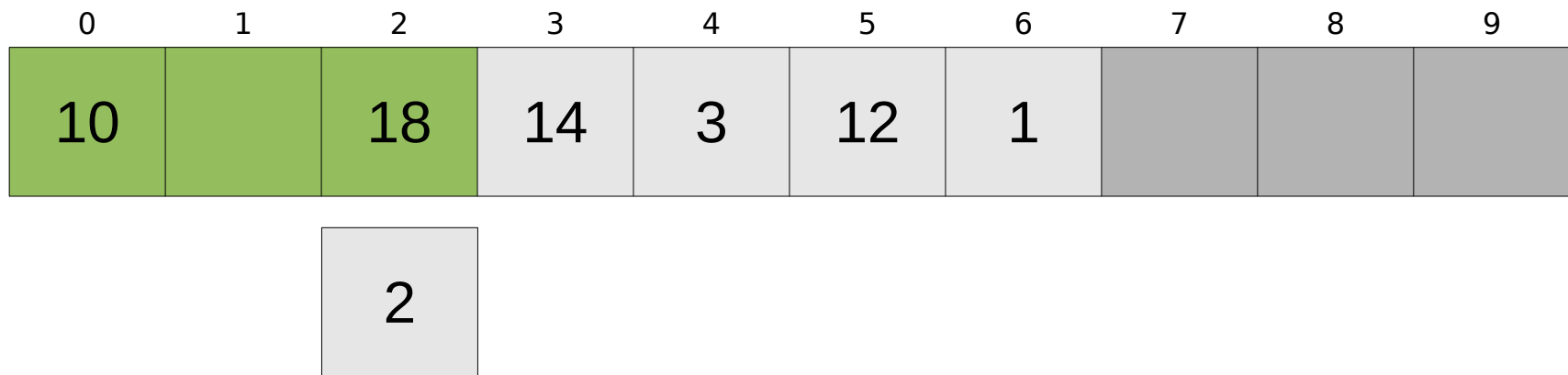


Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

# Insertion Sort

(insert each element into a sorted partition)



Pull the first element out of the **unsorted** partition of the vector.

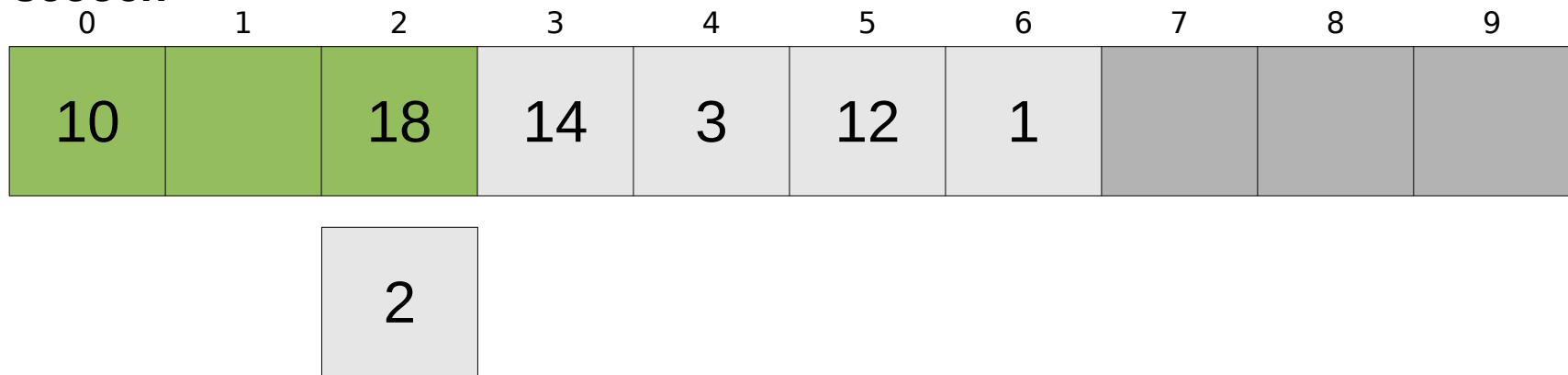
Start **scooching** things over if they're greater than 2.



# Insertion Sort

(insert each element into a sorted partition)

**\*scooch\***

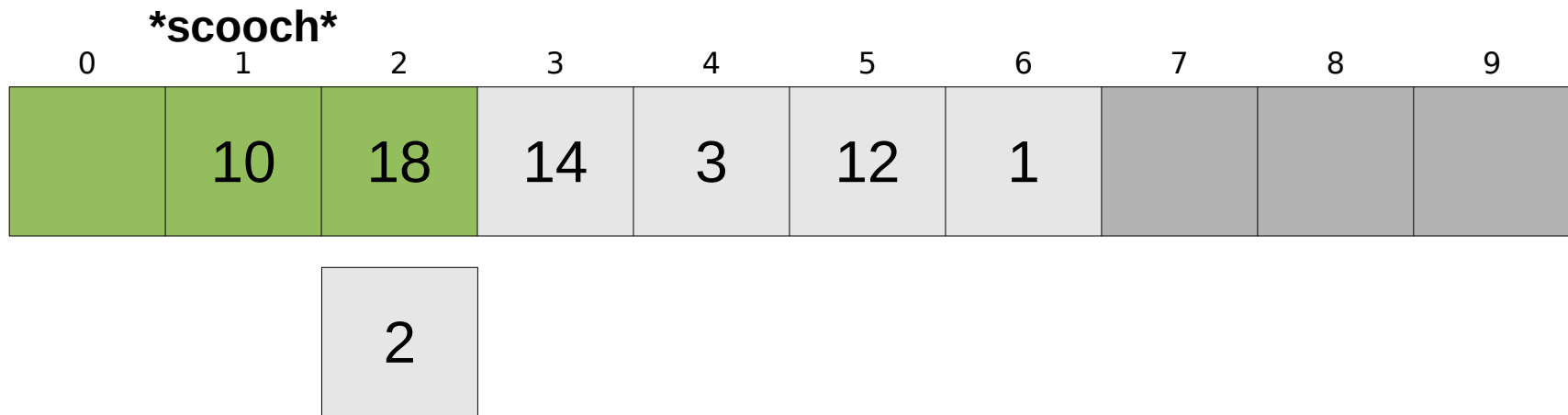


Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

# Insertion Sort

(insert each element into a sorted partition)

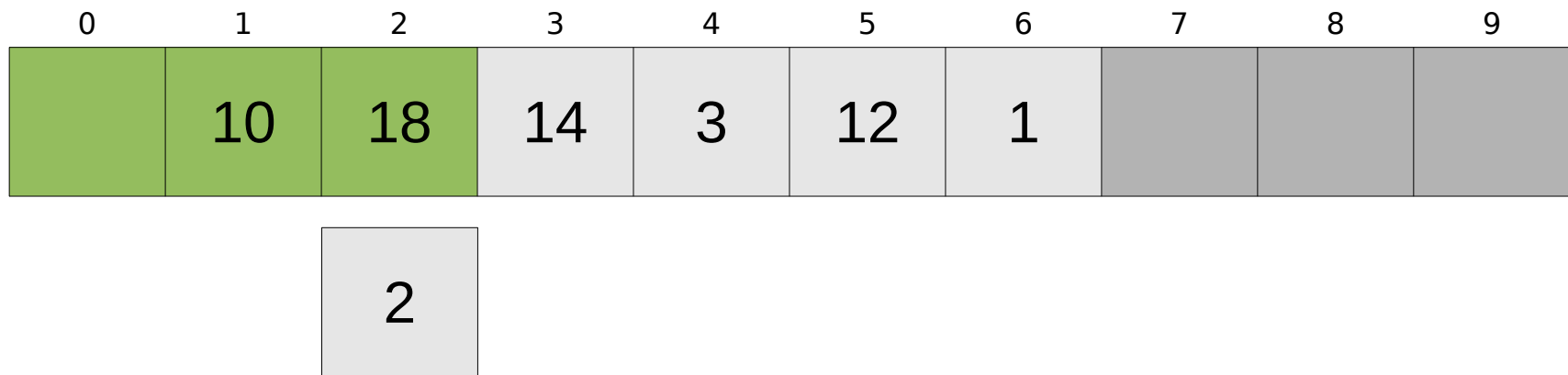


Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

# Insertion Sort

(insert each element into a sorted partition)

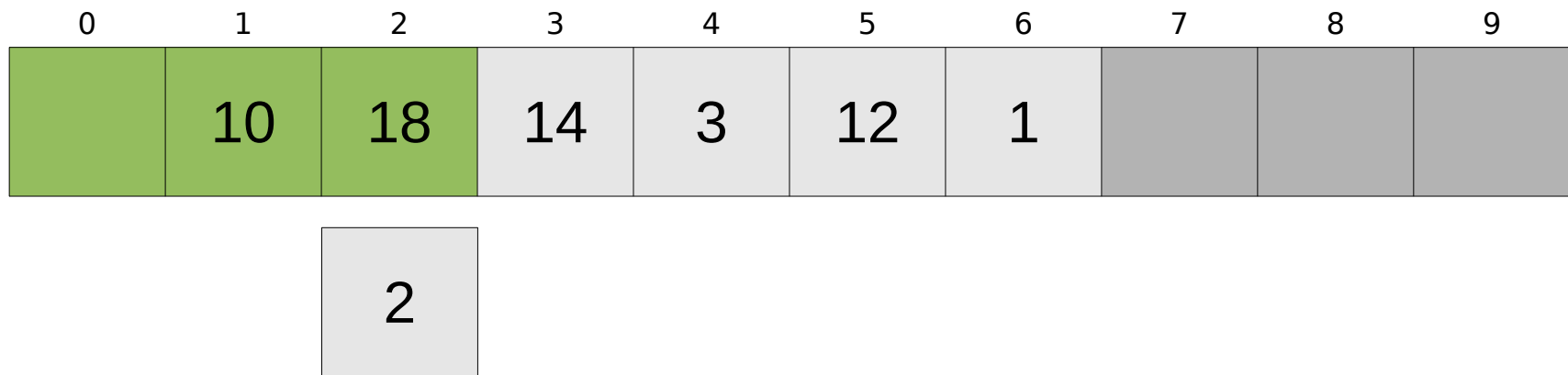


Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

# Insertion Sort

(insert each element into a sorted partition)



Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

Stick the element into the hole left for it in the **sorted** portion of the vector.

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
2	10	18	14	3	12	1			

Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

Stick the element into the hole left for it in the **sorted** portion of the vector.

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
2	10	18	14	3	12	1			

**\*TADA!\***

Pull the first element out of the **unsorted** partition of the vector.

Start **scooching** things over if they're greater than 2.

Stick the element into the hole left for it in the **sorted** portion of the vector.

# Insertion Sort

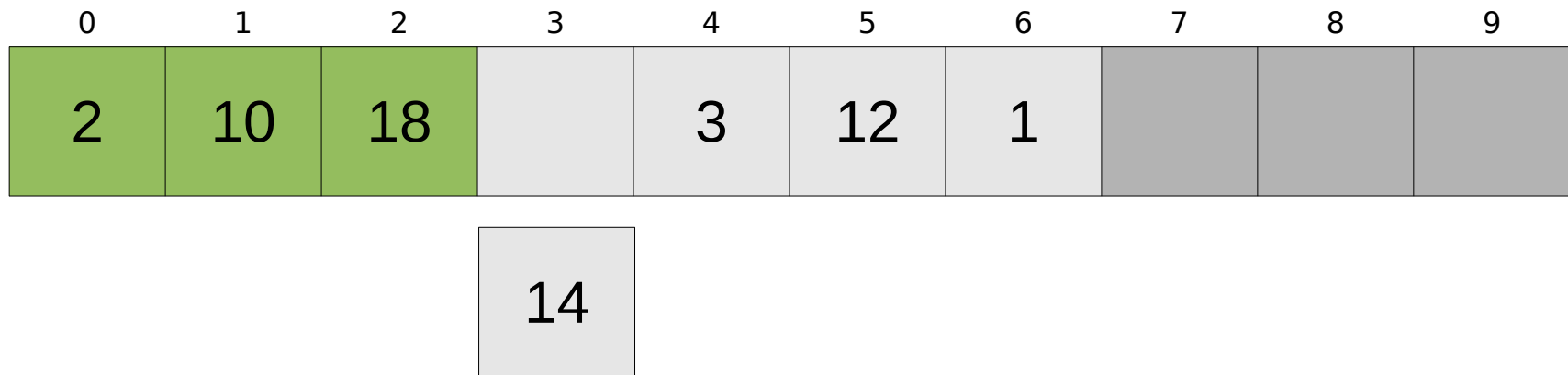
(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
2	10	18	14	3	12	1			

**See if you can follow along...**

# Insertion Sort

(insert each element into a sorted partition)

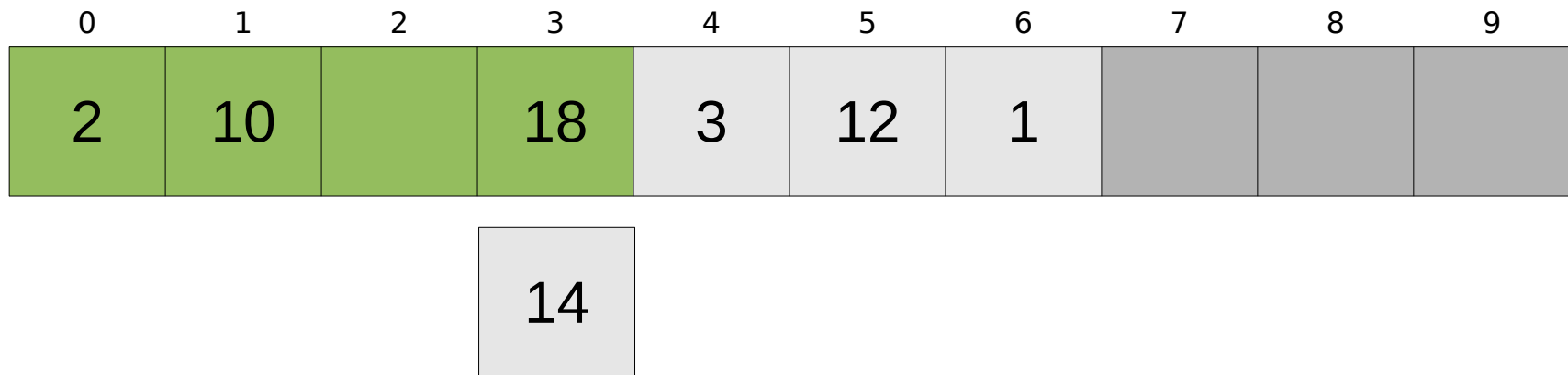


**See if you can follow along...**



# Insertion Sort

(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

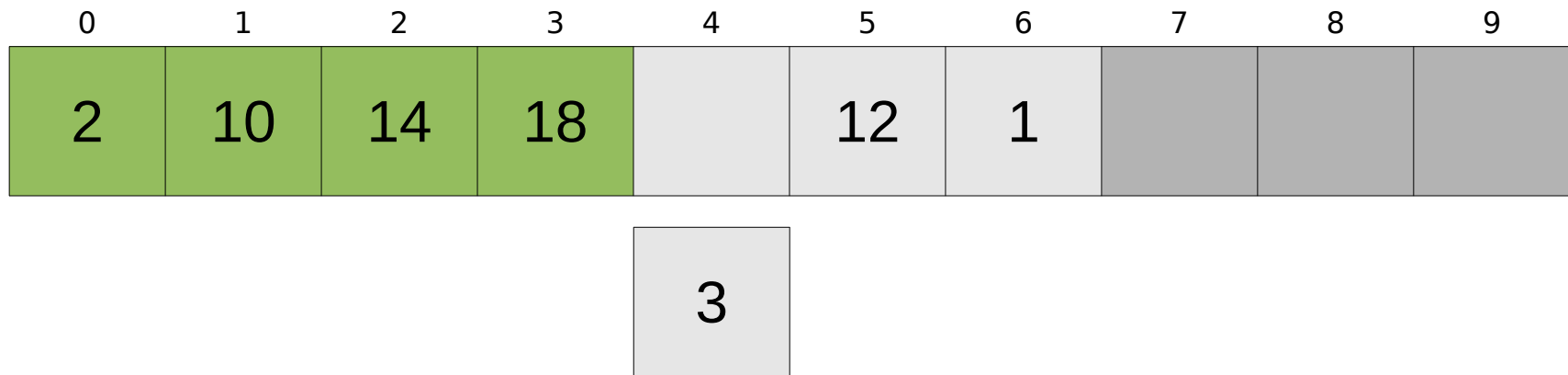
(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
2	10	14	18	3	12	1			

**See if you can follow along...**

# Insertion Sort

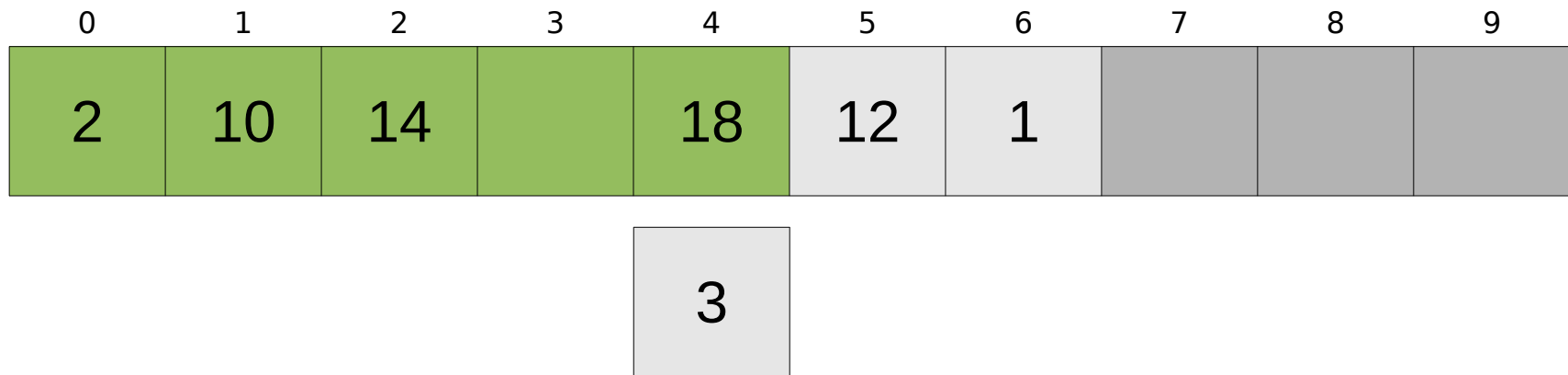
(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

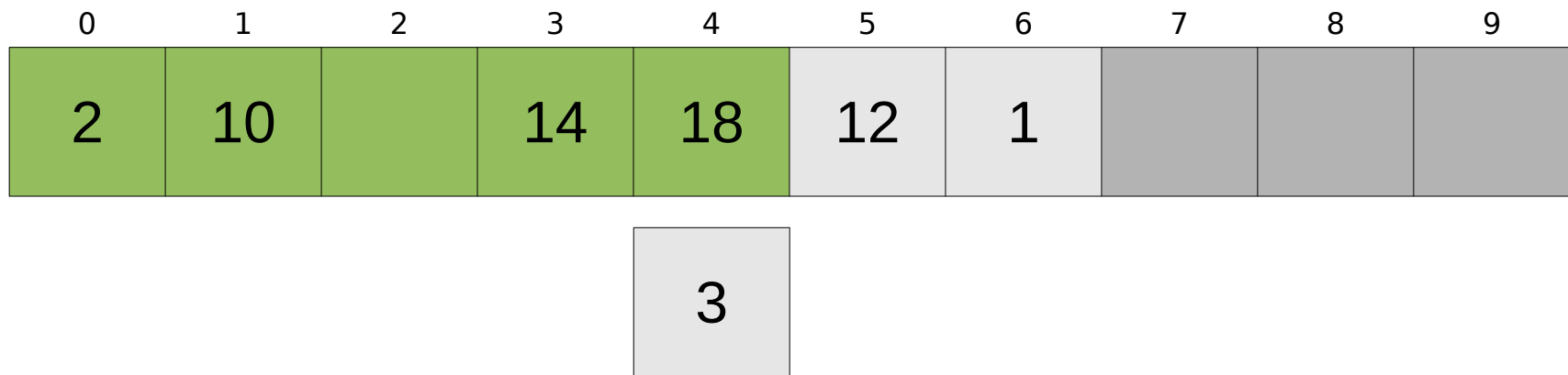
(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

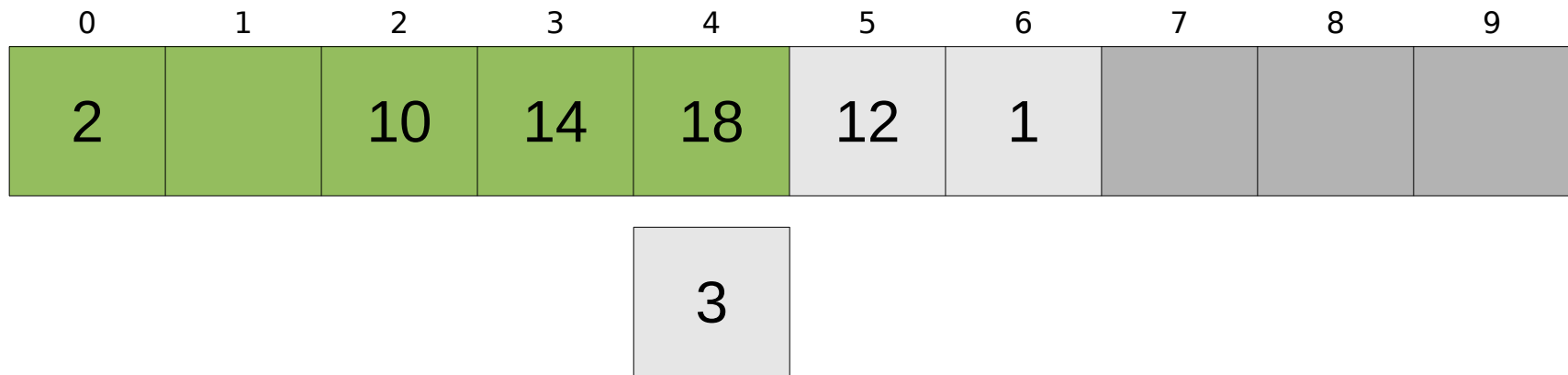
(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

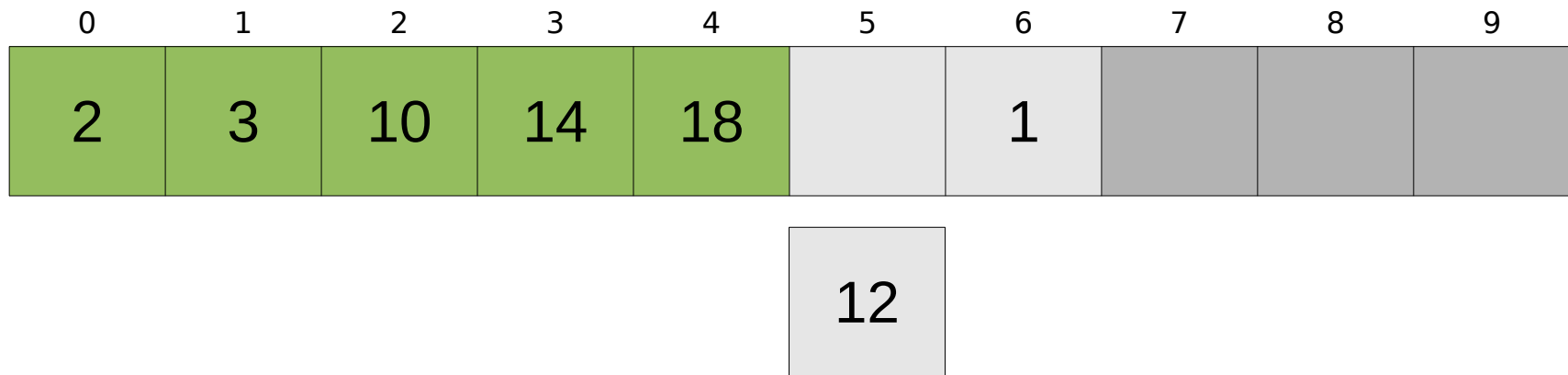
(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
2	3	10	14	18	12	1			

**See if you can follow along...**

# Insertion Sort

(insert each element into a sorted partition)

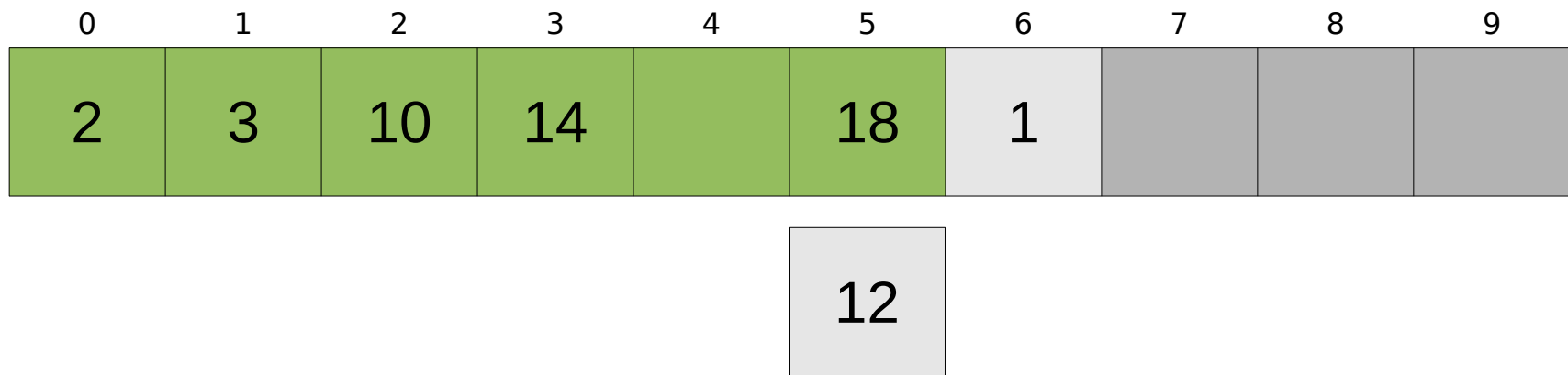


See if you can follow along...



# Insertion Sort

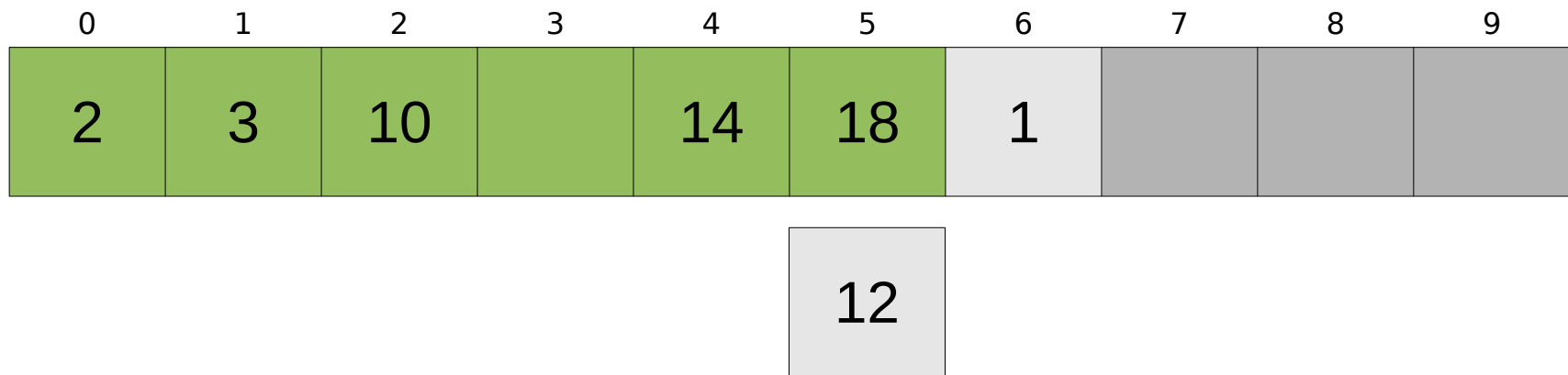
(insert each element into a sorted partition)



See if you can follow along...

# Insertion Sort

(insert each element into a sorted partition)



See if you can follow along...

# Insertion Sort

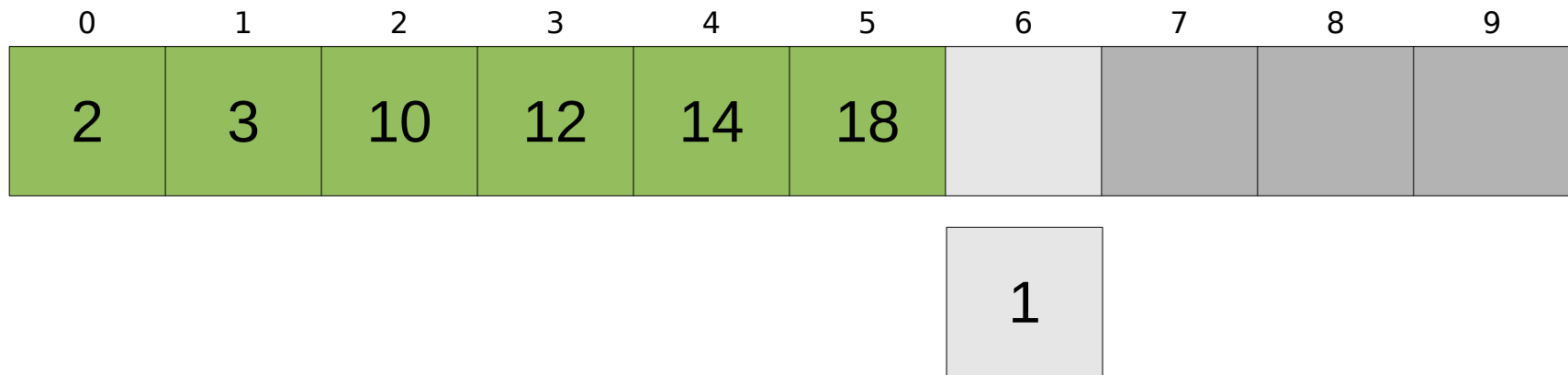
(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
2	3	10	12	14	18	1			

**See if you can follow along...**

# Insertion Sort

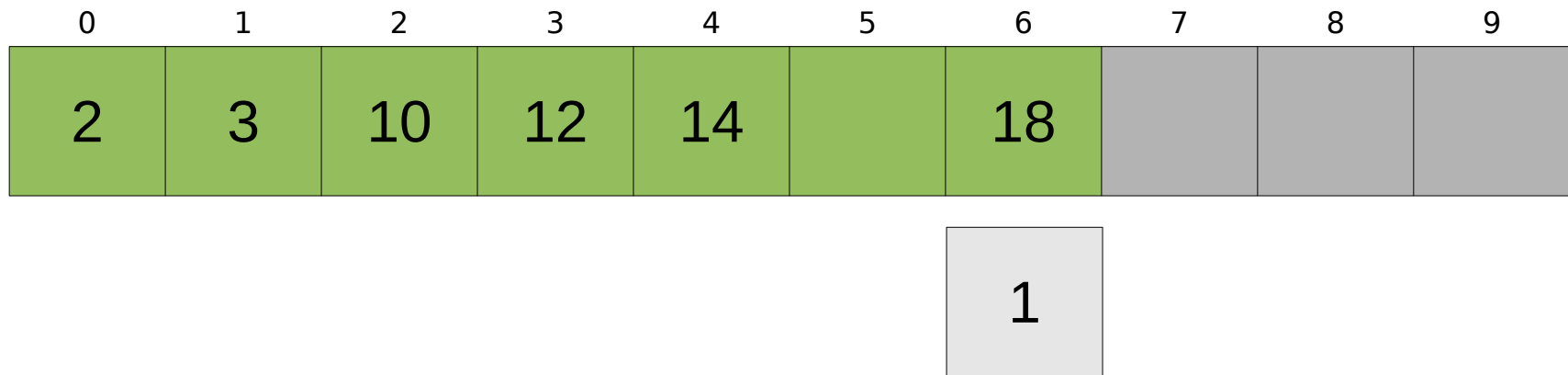
(insert each element into a sorted partition)



See if you can follow along...

# Insertion Sort

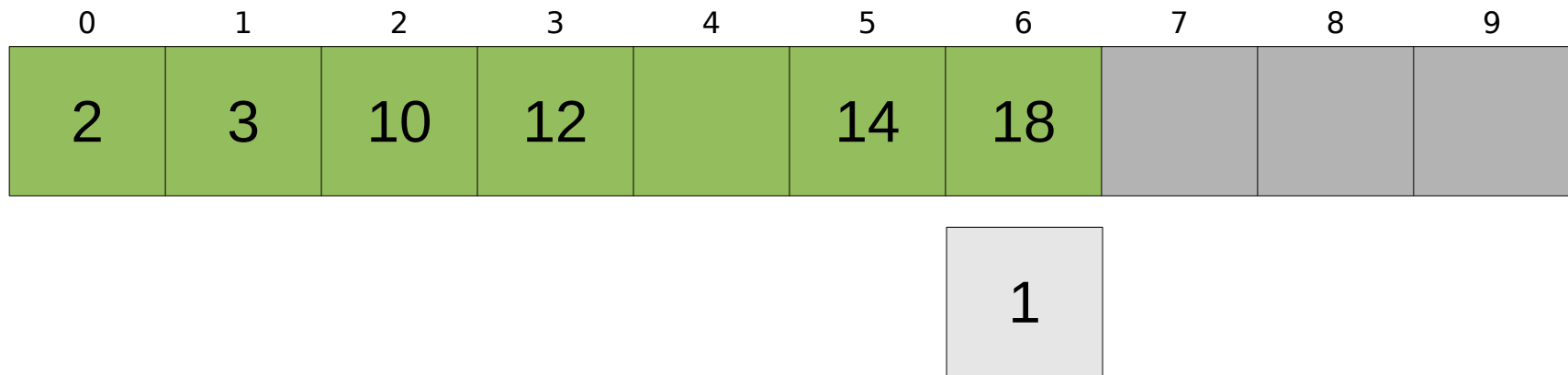
(insert each element into a sorted partition)



See if you can follow along...

# Insertion Sort

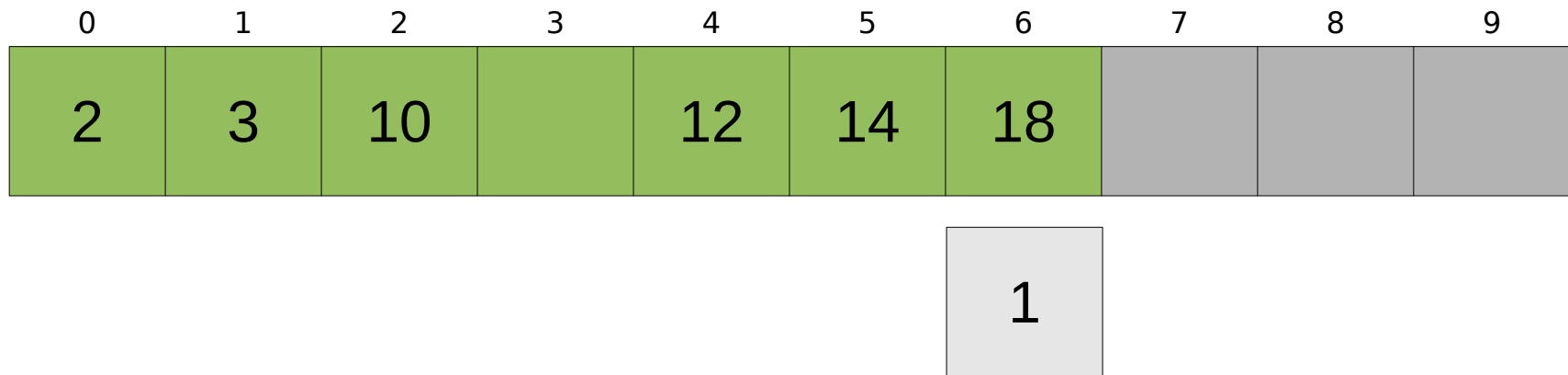
(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

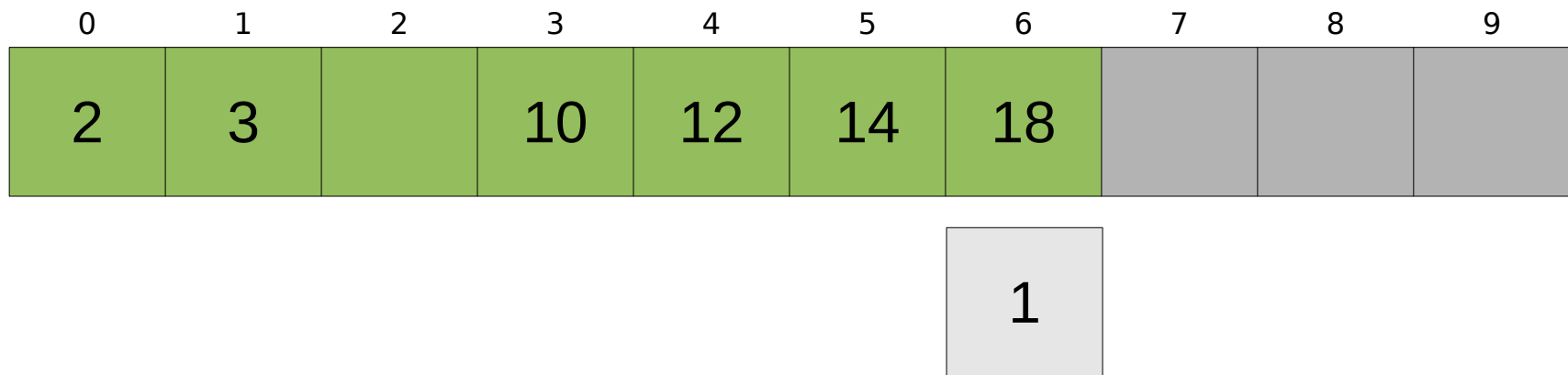
(insert each element into a sorted partition)



See if you can follow along...

# Insertion Sort

(insert each element into a sorted partition)

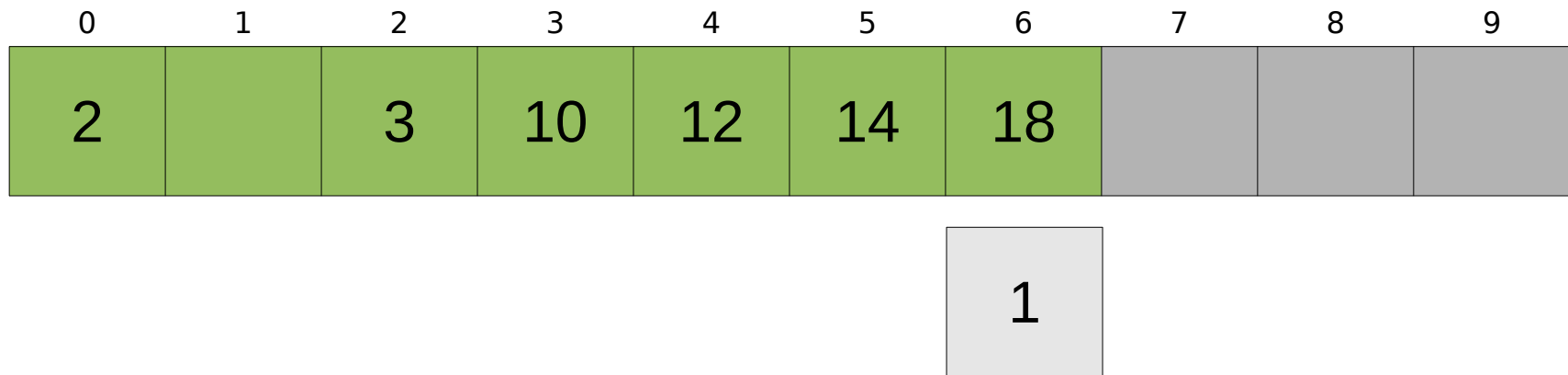


See if you can follow along...



# Insertion Sort

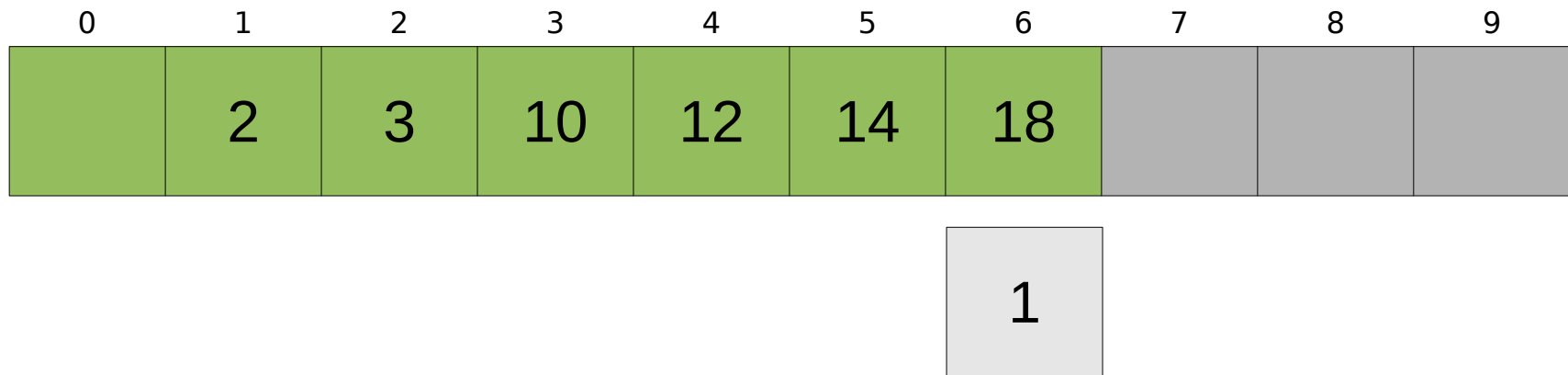
(insert each element into a sorted partition)



See if you can follow along...

# Insertion Sort

(insert each element into a sorted partition)



**See if you can follow along...**

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
1	2	3	10	12	14	18			

**We're finished!**

(Hooray!)

# Insertion Sort

(insert each element into a sorted partition)

0	1	2	3	4	5	6	7	8	9
1	2	3	10	12	14	18			

1. What's the **worst-case** Big-Oh runtime?
2. What's the **best-case** Big-Oh runtime?
3. Show the vector after **each pass** of Insertion Sort.

# Insertion Sort

(let's code it up)

10	18	2	14	3	12	1			
----	----	---	----	---	----	---	--	--	--

```
void insertionSort(Vector<int>& v)
{
    for (int i = 1; i < v.size(); i++)
    {
        int val = v[i];
        int gap = i;

        for (int j = gap - 1; j >= 0 && v[j] > val; j--)
        {
            v[j + 1] = v[j];
            gap--;
        }

        v[gap] = val;
    }
}
```