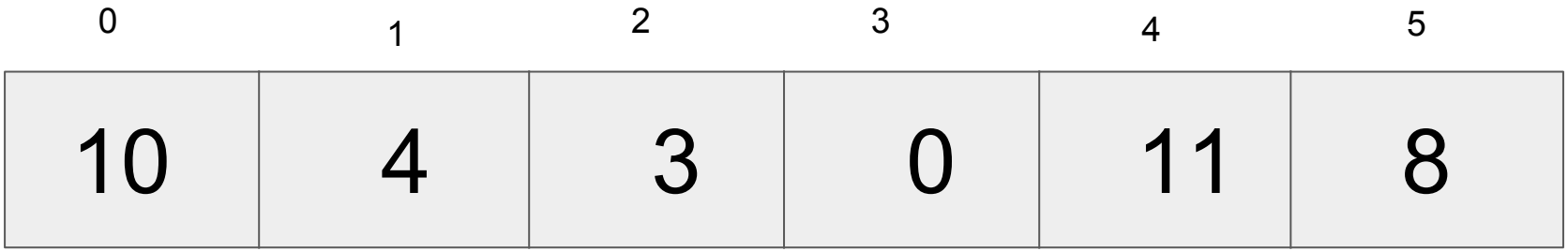
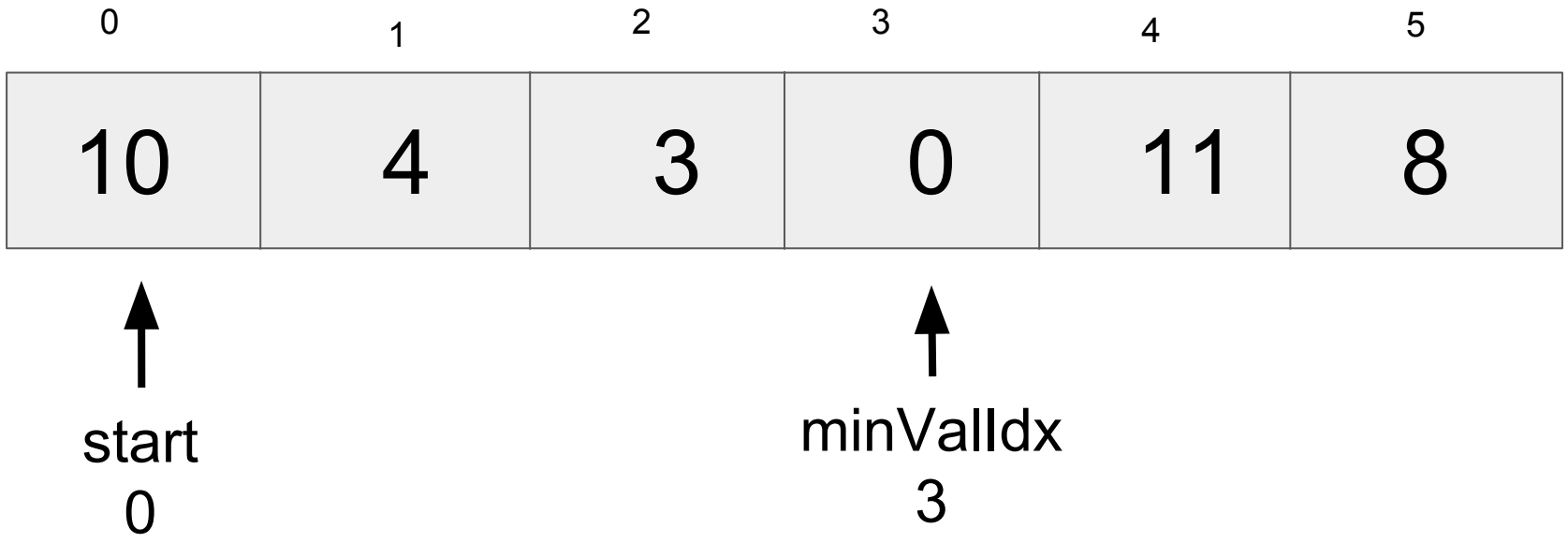
Selection Sort

Selection Sort



What do we do first?

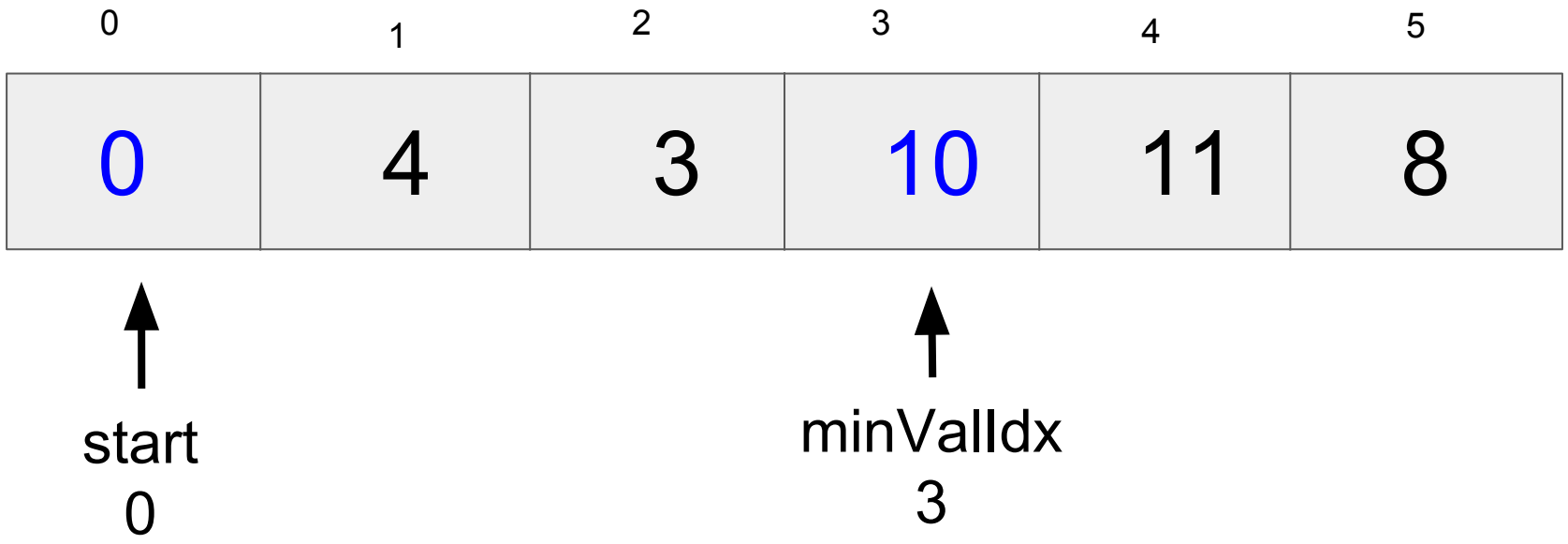
Selection Sort



Find minimum element idx between start to end

What next?

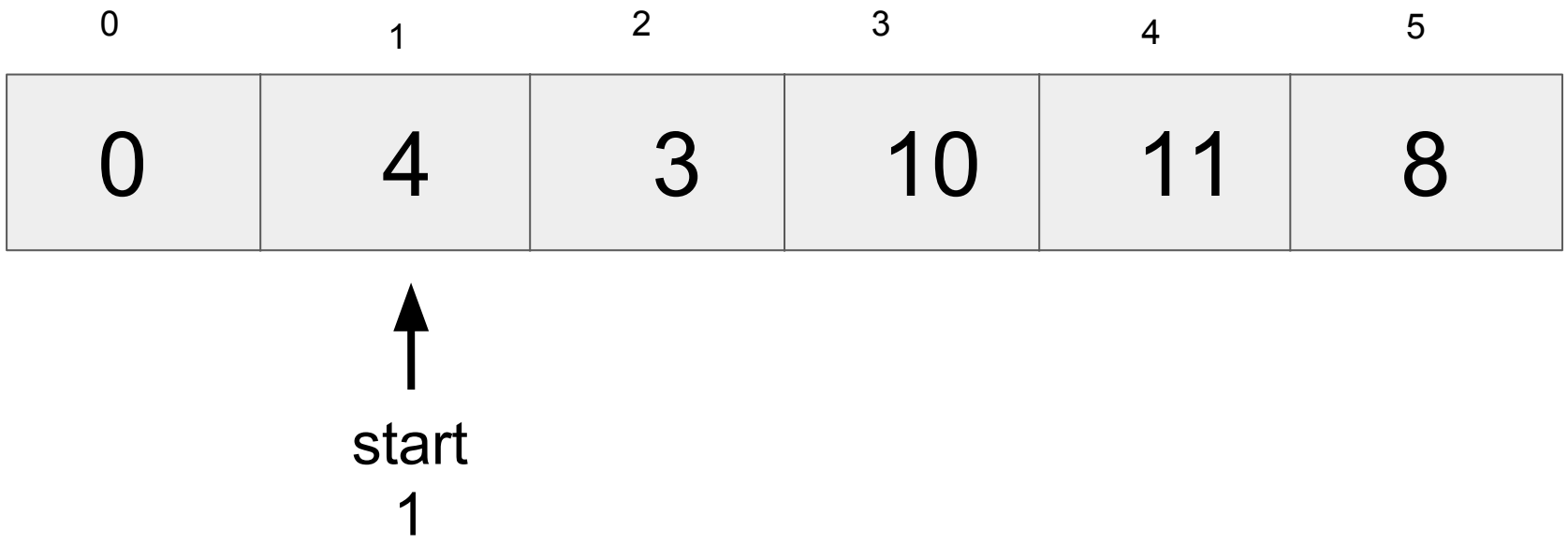
Selection Sort



Swap the elements at start and minValIdx

What next?

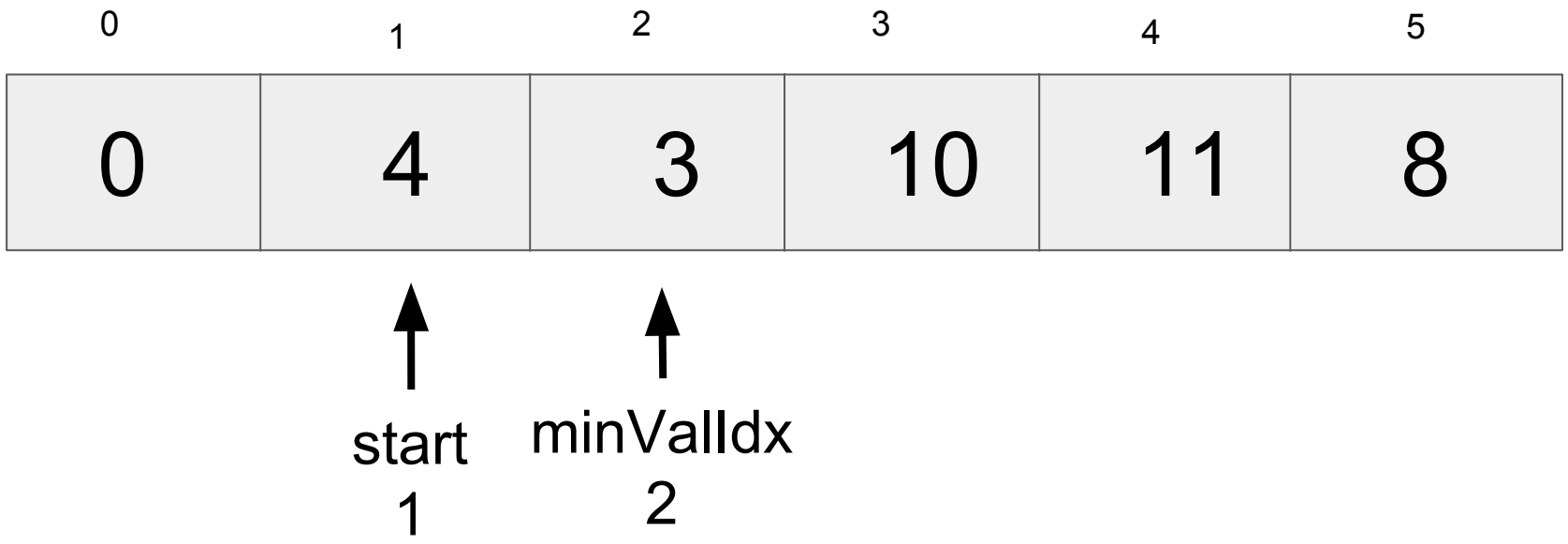
Selection Sort



Decrease the interval.

What next?

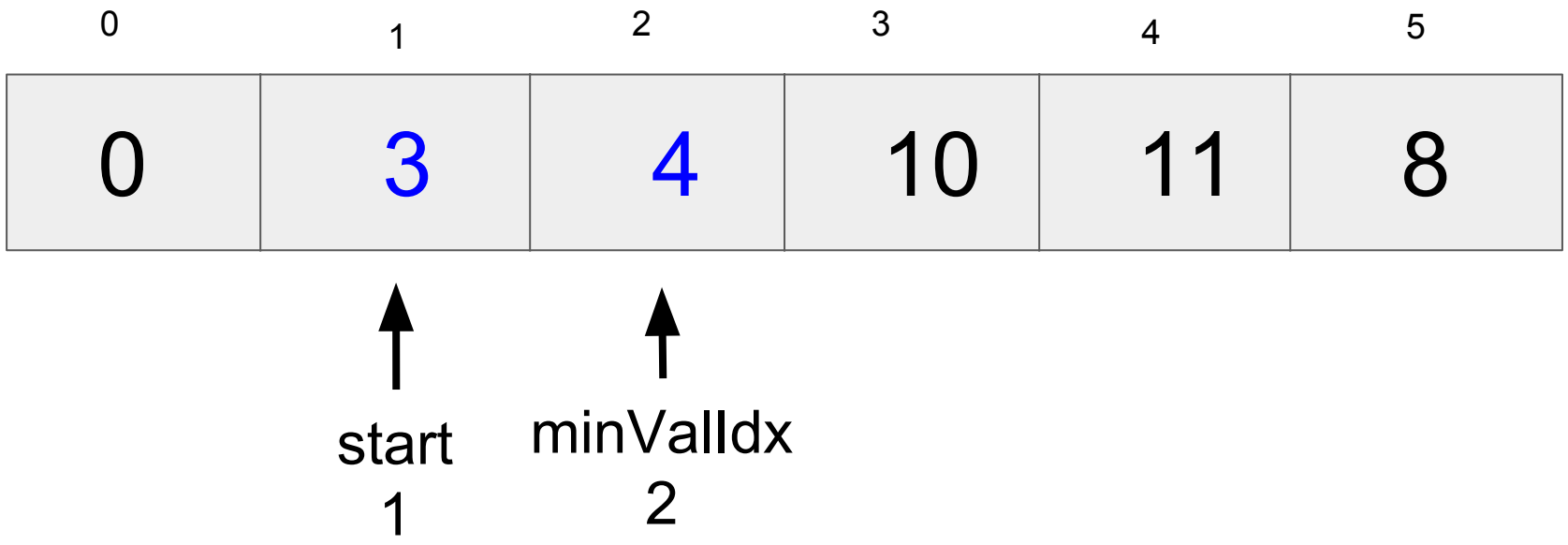
Selection Sort



Find minimum element between start to end

What next?

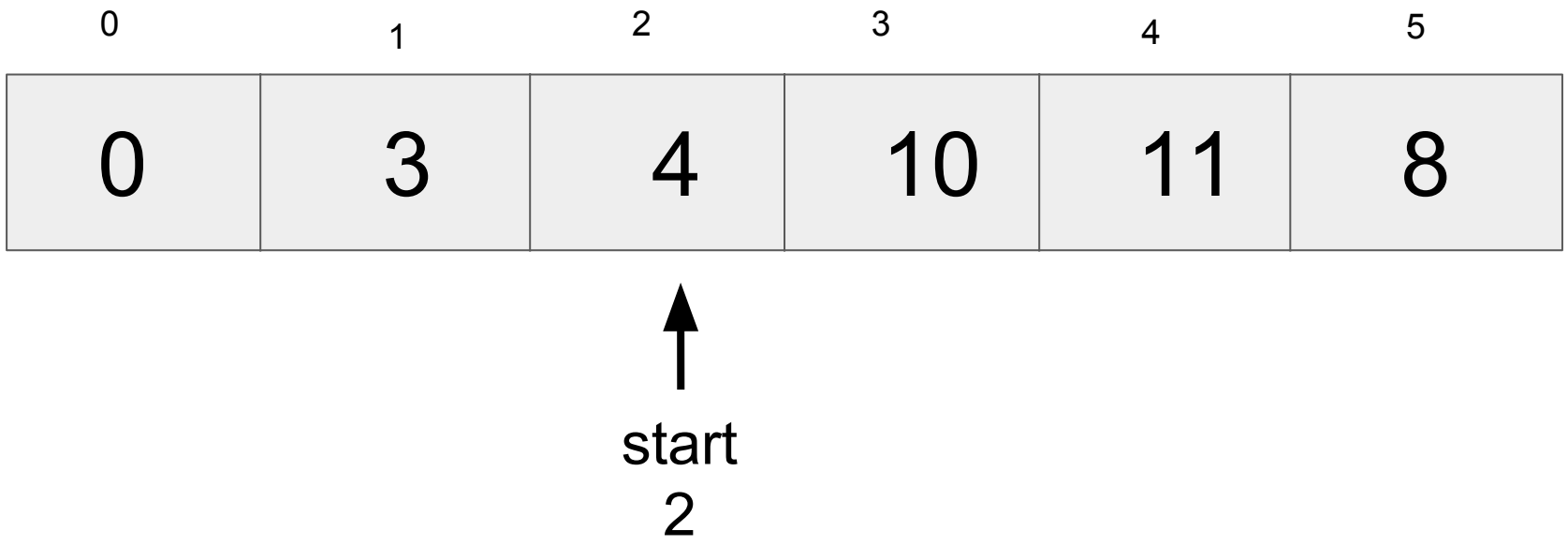
Selection Sort



Swap the elements at start and minValIdx

What next?

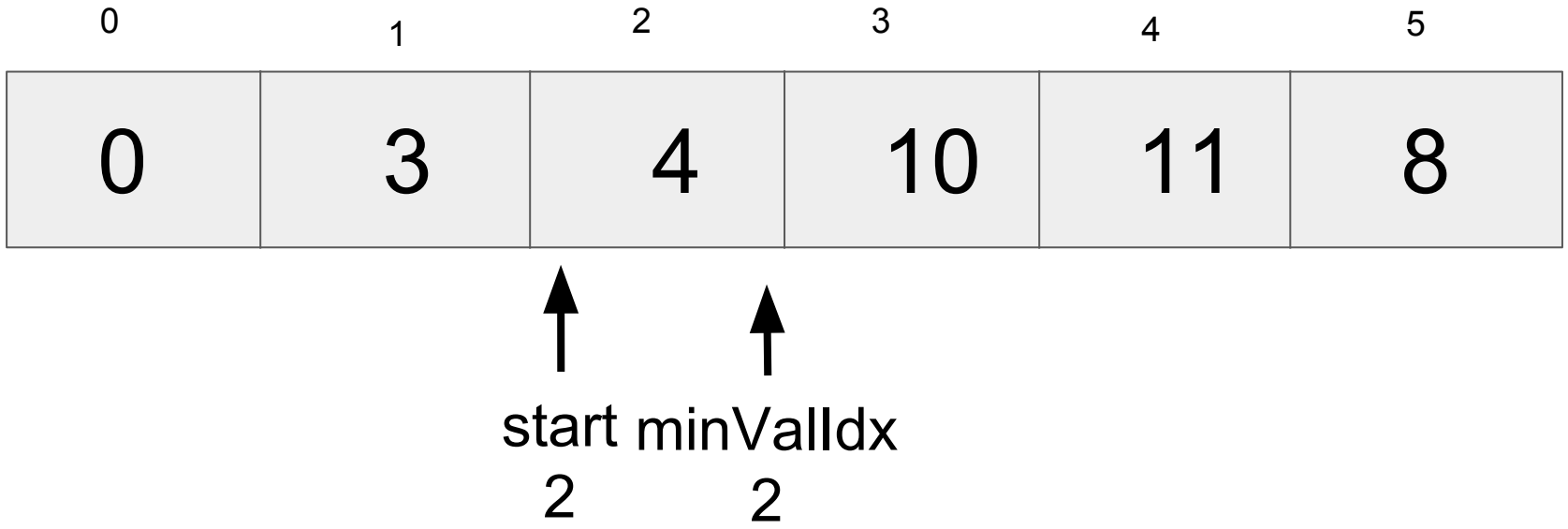
Selection Sort



Decrease the interval.

What next?

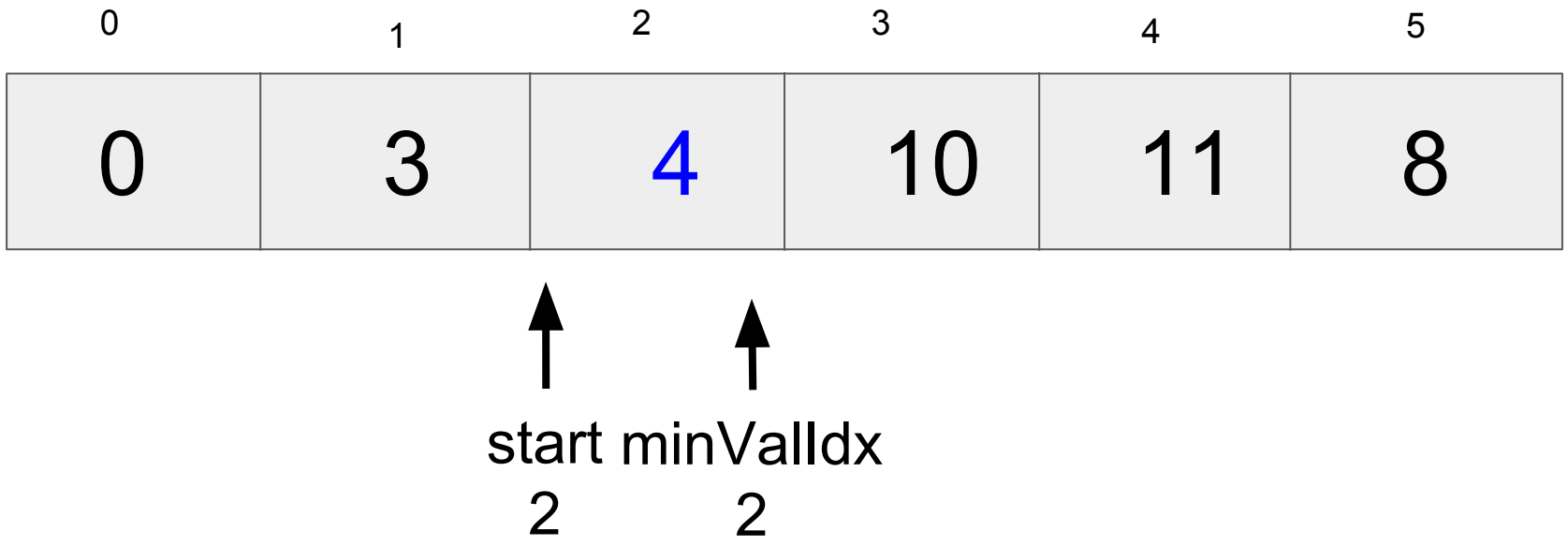
Selection Sort



Find minimum element idx between start to end

What next?

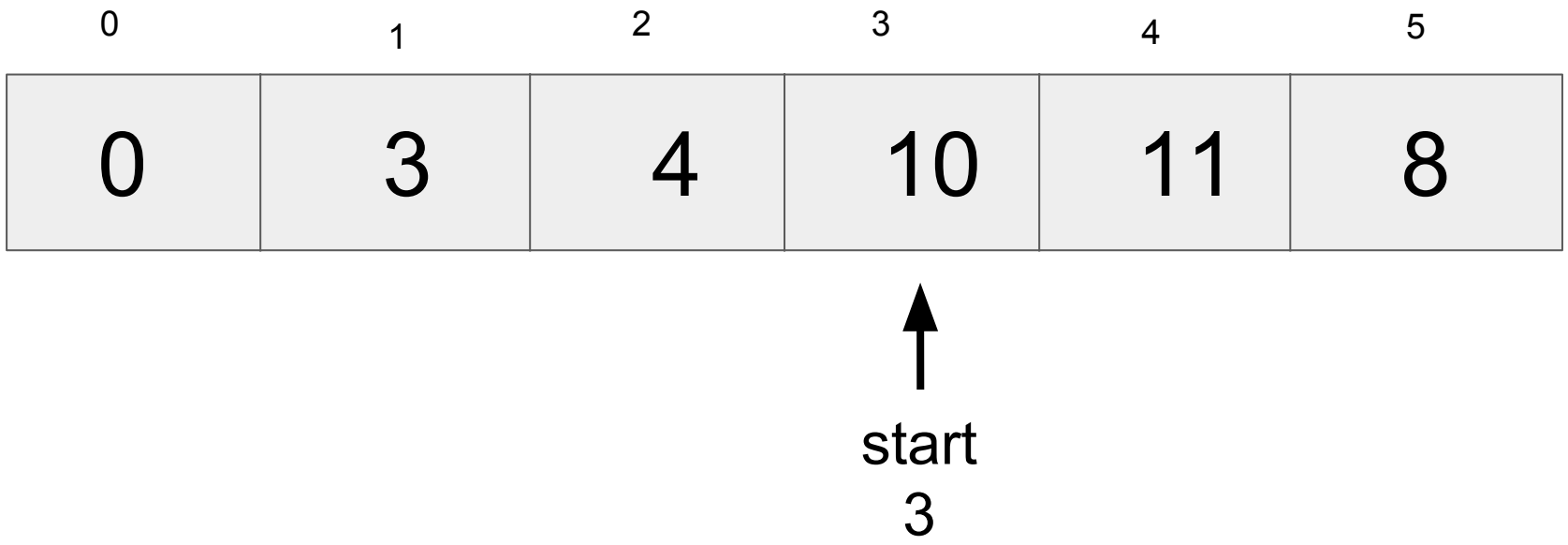
Selection Sort



Swap the elements at start and minValIdx

What next?

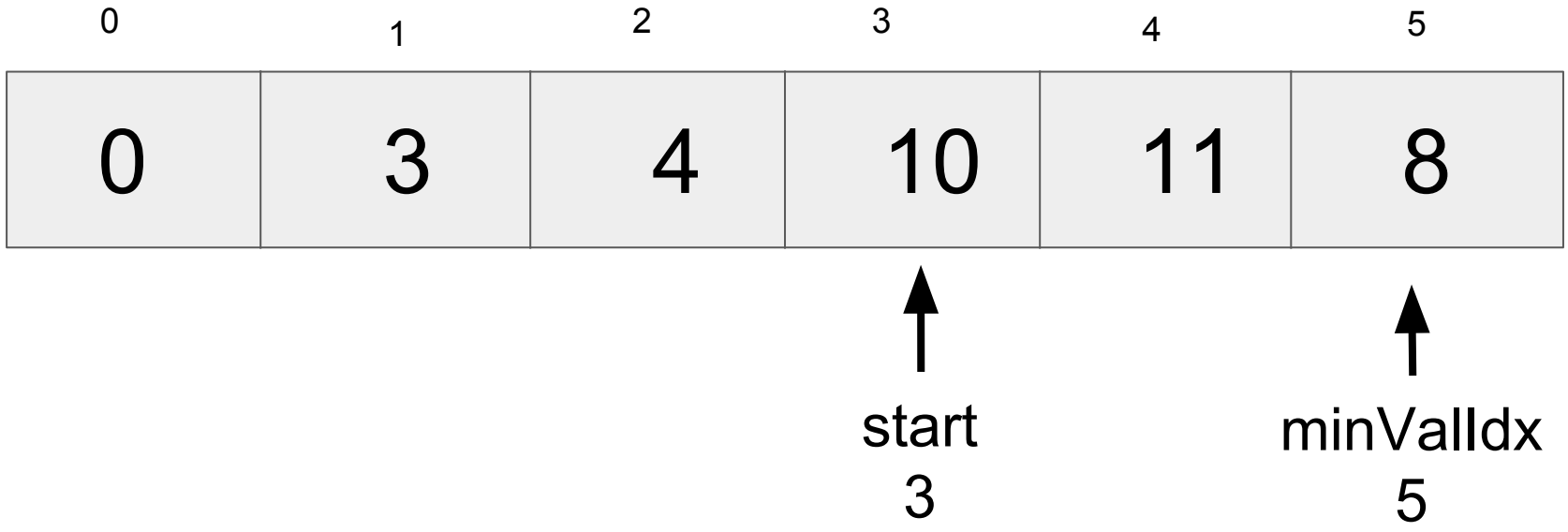
Selection Sort



Decrease the interval.

What next?

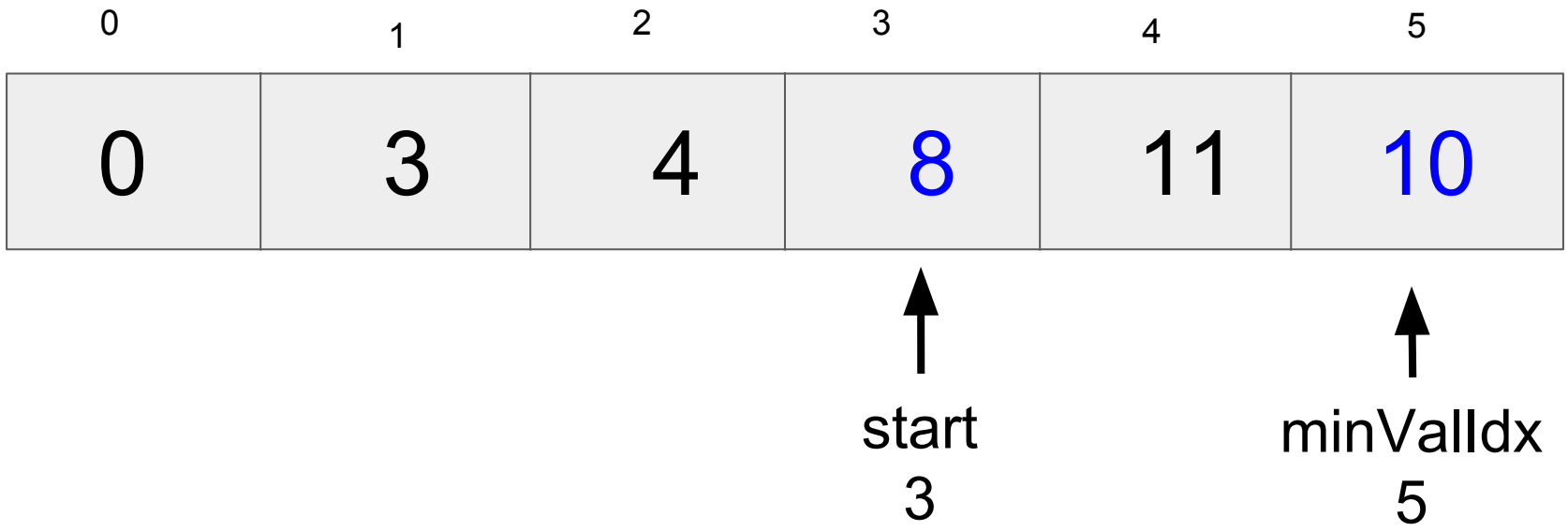
Selection Sort



Find minimum element idx between start to end

What next?

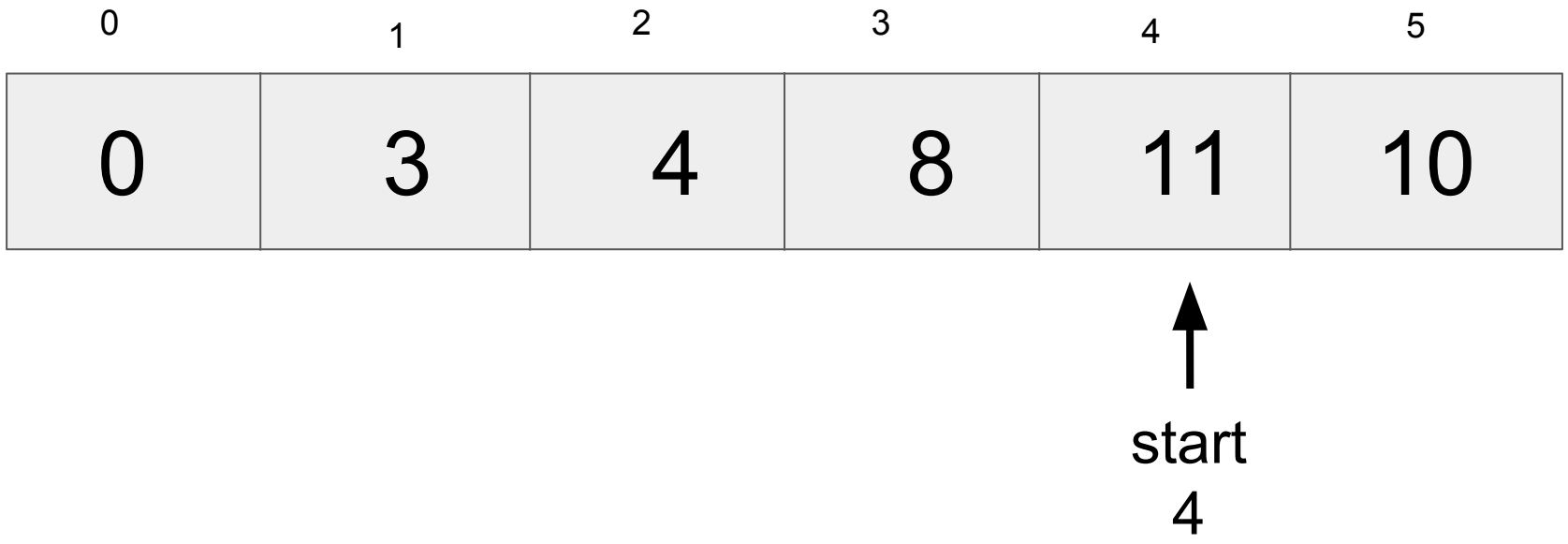
Selection Sort



Swap the elements at start and minValIdx

What next?

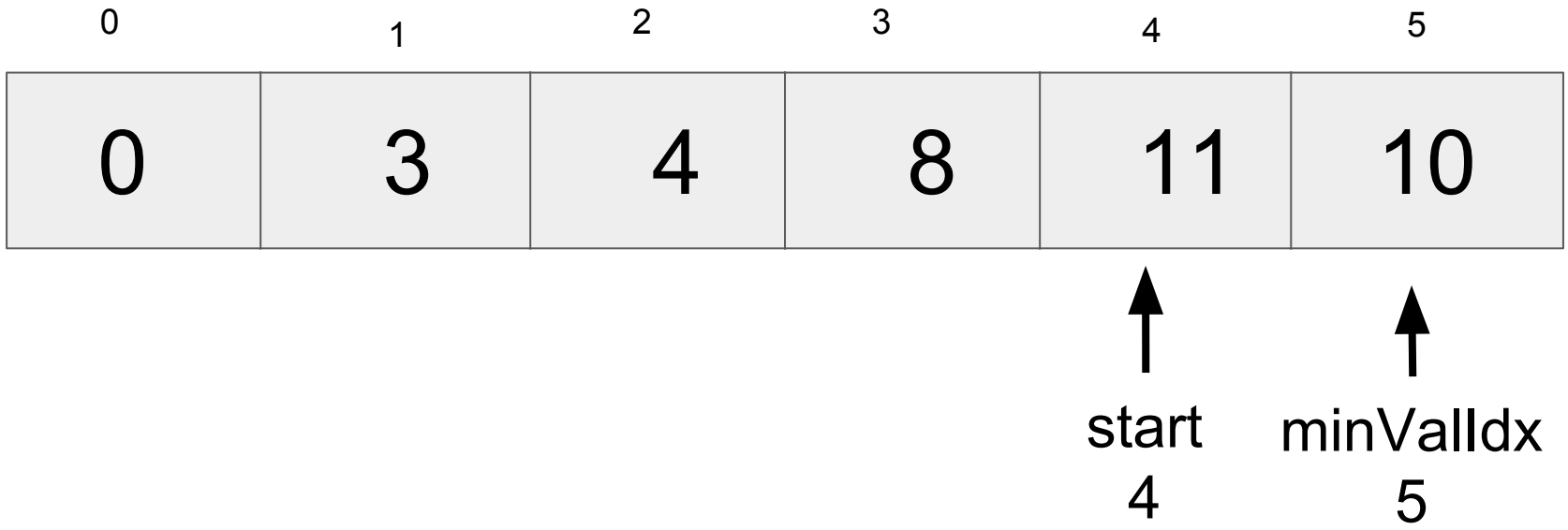
Selection Sort



Decrease the interval.

What next?

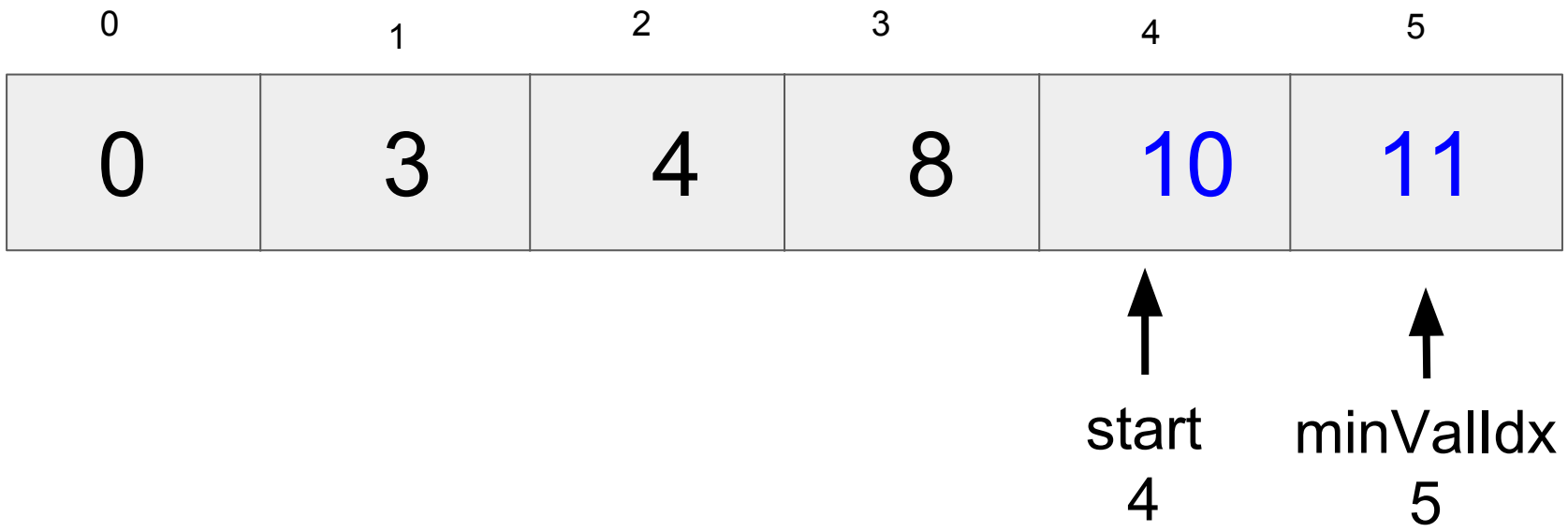
Selection Sort



Find minimum element idx between start to end

What next?

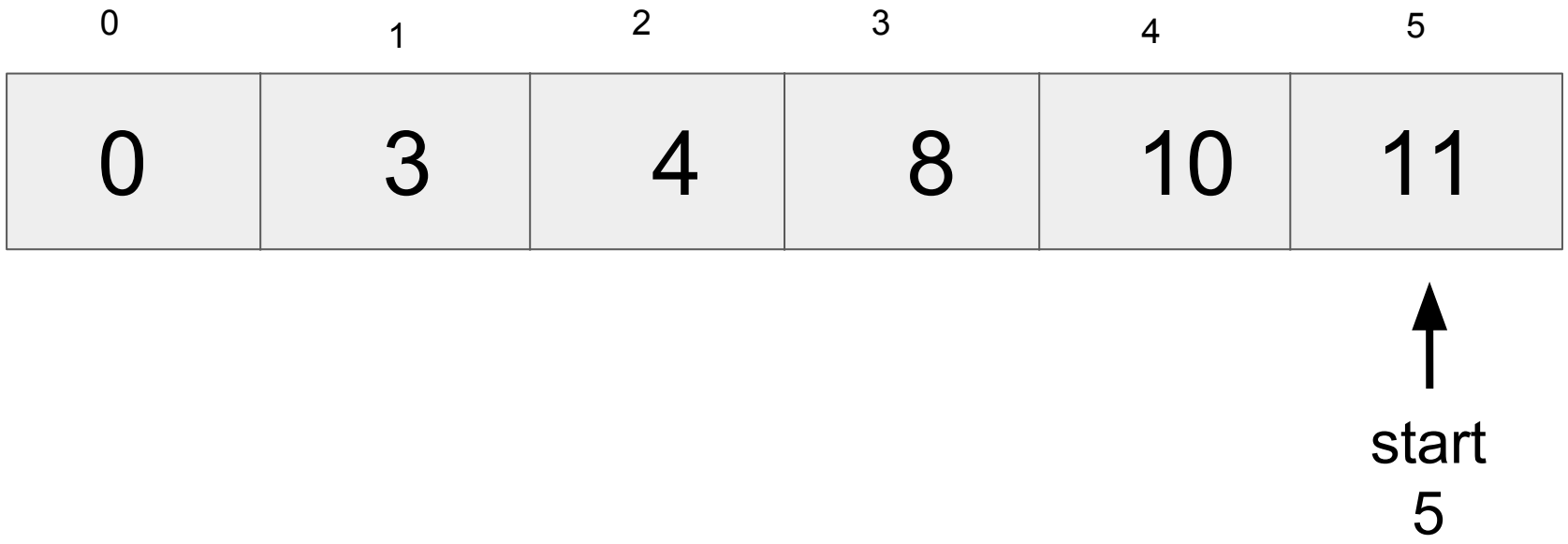
Selection Sort



Swap the elements at start and minValIdx

What next?

Selection Sort



Decrease the interval.

We’re done!

Selection sort

```
selectionSort(L):
    for startIdx in range(len(L)):
        minIdx = findMinimum(startIdx, L)
        swap(startIdx, minIdx, L)
```

Selection sort

```
findMinimum(startIdx, L):
    minIdx = startIdx
    for i in range(startIdx, len(L)):
        if L[i] < L[minIdx]:
            minIdx = i
    return minIdx
```

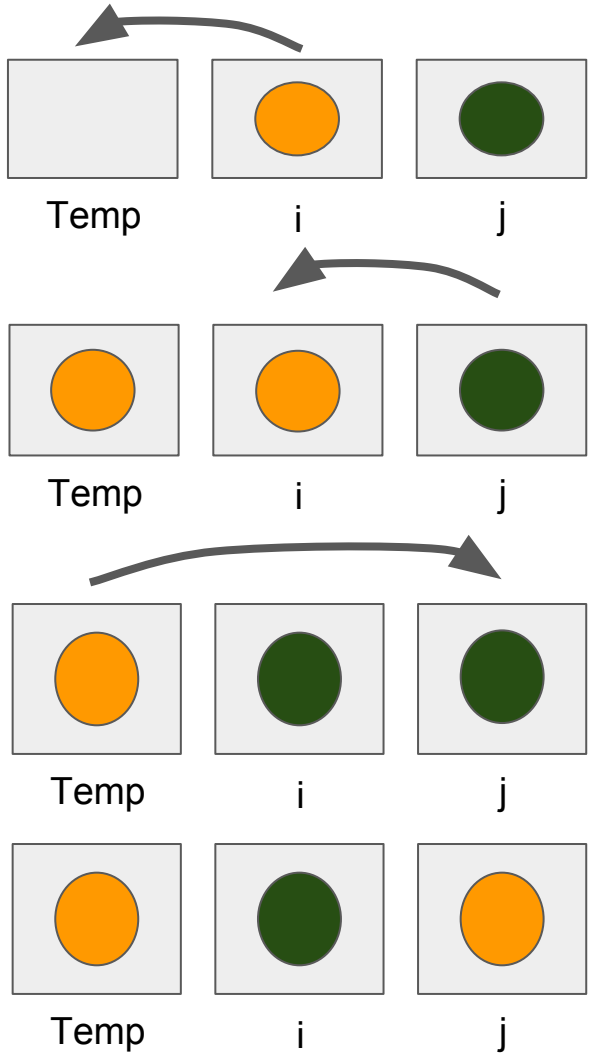
Selection sort

swap(i, j, L):

```
    temp = L[i] # step 1

    L[i] = L[j]    # step 2

    L[j] = temp # step 3
```



swap.py — ~/classes/cs21/f18/library.git/inclass/w10 — Atom

File Edit View Selection Find Packages Help

swap.py | isSorted.py | selectionSort.py | sort-sim.py | addEx2.py | useAdd.py | addEx1.py

```
"""
Write a function that swaps two elements

NOTE:
 The if statement at the bottom only executes if we run this
 file from the command line, e.g.

 >> python3 swap.py
 Before: [0, 1]
 After: [1, 0]

 This feature allows us to use the functions in this file in other
 programs using the syntax

 import swap
 swap.swap(0,1,L)

"""

def swap(i, j, L):
    """
    Swaps the ith and jth elements of L
    Params i,j (int): indexes into L
    Param L (list): the list to change
    Returns: None
    """
    tmp = L[j]
    L[j] = L[i]
    L[i] = tmp


if __name__ == '__main__':
   L = [0,1]
   print("Before:", L)
   swap(0,1,L)
   print("After:", L)
```

swap.py 1:1 LF N UTF-8 Python updates Fetch 23 files

isSorted.py — ~/classes/cs21/f18/library.git/inclass/w10 — Atom

File Edit View Selection Find Packages Help

swap.py | isSorted.py | selectionSort.py | sort-sim.py | addEx2.py | useAdd.py | addEx1.py

```
"""
Write a function that checks if a list is sorted from least to greatest

  $ python3 isSorted.py
  [0, 2, 4, 6] isSorted? True
  [6, 2, 4, 0] isSorted? False
  [10] isSorted? True

"""

import random

def isSorted(L):
    """
    Returns True if the list L is sorted; False otherwise
    Param L (list): the list to test
    Return (bool)
    """
    for i in range(len(L)-1):
        if L[i] > L[i+1]:
            return False
    return True

if __name__ == '__main__':

   L = list(range(0,8,2))
   print(L, "isSorted?", isSorted(L))

   random.shuffle(L)
   print(L, "isSorted?", isSorted(L))

   L = [10]
   print(L, "isSorted?", isSorted(L))

```

isSorted.py 12:1 LF N UTF-8 Python updates Fetch 23 files

selectionSort.py — ~/classes/cs21/f18/library.git/inclass/w10 — Atom

File Edit View Selection Find Packages Help

swap.py | isSorted.py | selectionSort.py | sort-sim.py | addEx2.py | useAdd.py | addEx1.py

```
"""
Sort a list in place using selection sort.
Use your existing implementation for swap and isSorted!

 $ python3 selectionSort.py
 Before: [10, 4, 3, 0, 11, 8]
 swap 0 3
 swap 1 2
 swap 2 2
 swap 3 5
 swap 4 5
 swap 5 5
 After: [0, 3, 4, 8, 10, 11] IsSorted? True

"""

import swap
import random
import isSorted

def FindMinimumIdx(start, L):
    minVal = L[start]
    minIdx = start
    for i in range(start, len(L)):
        if L[i] < minVal:
            minVal = L[i]
            minIdx = i

    return minIdx

def selectionSort(L):
    """
    Sort the list L in place using selection sort
    Param L (list): the list to sort
    Return: None
    """
    for start in range(len(L)):
        minVal = L[start]
        minIdx = start
        for i in range(start, len(L)):
            if L[i] < minVal:
                minVal = L[i]
                minIdx = i
        swap.swap(minIdx, start, L)
        print("swap", start, minIdx)

if __name__ == '__main__':
   L = [10,4,3,0,11,8]
   print("Before:", L)
   selectionSort(L)
   print("After:", L, "IsSorted?", isSorted.isSorted(L))
```

selectionSort.py 2:27 LF N UTF-8 Python updates Fetch 23 files