

# 159 XTENDED

## A WEEKLY REVIEW

### SORTING ALGORITHMS

Pull up [Socrative.com](https://Socrative.com)

and join room **XTEND159**



## A CLASSIC PROBLEM IN COMPUTER SCIENCE.

- ▶ How can we take an unordered list of numbers...

15, 8, 190, 252, 0, -9, 3001, 1, 99

and put them in order?

-9, 0, 1, 8, 15, 99, 190, 252, 3001

## SELECTION SORT

- ▶ Visual Sorting Algorithms

## SELECTION SORT

- When performing Selection Sort on the following list:

5, 18, 11, 49, 15, 5, 22, 10, 47, 39, 50, 13, 5, 1

What are the **second** pair of numbers to be swapped?

Pull up **Socrative.com**

and join room **XTEND159**

# WHAT IS THE EFFICIENCY OF SELECTION SORT?

```
public static void selectionSort( int[] numbers ) {  
    int min;  
    int temp;  
  
    for( int index = 0; index < numbers.length - 1; index++ ) {  
        min = index;  
  
        for( int scan = index + 1; scan < numbers.length; scan++ ) {  
            if( numbers[ scan ] < numbers[ min ] ) {  
                min = scan;  
            }  
        }  
  
        // swap the values  
        temp = numbers[ min ];  
        numbers[ min ] = numbers[ index ];  
        numbers[ index ] = temp;  
    }  
}
```

Pull up **Socrative.com**  
and join room **XTEND159**

## WHAT IS THE EFFICIENCY OF SELECTION SORT?

```
public static void selectionSort( int[] numbers ) {  
    int min;  
    int temp;
```

```
        for( int index = 0; index < numbers.length - 1; index++ ) {  
            min = index;
```

```
                for( int scan = index + 1; scan < numbers.length; scan++ ) {  
                    if( numbers[ scan ] < numbers[ min ] ) {  
                        min = scan;  
                    }  
                }
```

```
            // swap the values temp = numbers[ min ];  
            numbers[ min ] = numbers[ index ];  
            numbers[ index ] = temp;  
        }
```

$n$

$n$

$= n^2$

## INSERTION SORT

- ▶ Visual Sorting Algorithms

## INSERTION

- ▶ When performing Insertion Sort on the following list:

21, 9, 18, 17, 16, 43, 21, 35, 38, 13, 35, 24

The first operation performed is a swap of 21 and 9.

**What is the second operation performed?**

Pull up **Socrative.com**

and join room **XTEND159**

# WHAT IS THE EFFICIENCY OF INSERTION SORT?

```
public static void insertionSort( int[] numbers ) {  
  
    for( int index = 1; index < numbers.length - 1; index++ ) {  
        int key = numbers[index];  
        int position = index;  
  
        // shift larger values to the right  
        while( position > 0 && numbers[position - 1] > key ) {  
            numbers[position] = numbers[position - 1];  
            position--;  
        }  
  
        numbers[position] = key;  
    }  
}
```

Pull up **Socrative.com**  
and join room **XTEND159**

## WHAT IS THE EFFICIENCY OF INSERTION SORT?

```
public static void insertionSort( int[] numbers ) {  
    for( int index = 1; index < numbers.length - 1; index++ ) {  
        int key = numbers[index];  
        int position = index;  
  
        // shift larger values to the right  
        while( position > 0 && numbers[position - 1] > key ) {  
            numbers[position] = numbers[position - 1];  
            position--;  
        }  
  
        numbers[position] = key;  
    }  
}
```

 $n$  $= n^2$

## QUICK SORT

- ▶ Visual Sorting Algorithms

```
private int partition ( int[] numbers, int start, int end ) {  
    int pivotValue = numbers[ start ];  
    int pivotIndex = start;  
    int temp;  
    for ( int index = start + 1; index <= end; index++ ) {  
        if (numbers[index] < pivotValue) {  
            pivotIndex++;  
            temp = numbers[ pivotIndex ];  
            numbers[ pivotIndex ] = numbers[ index ];  
            numbers[ index ] = temp;  
        } // end if  
    } // end for  
  
    // switch pivot value into proper position  
    temp = numbers[start];  
    numbers[ start ] = numbers[ pivotIndex ];  
    numbers[ pivotIndex ] = temp;  
  
    return pivotIndex;  
}
```

Pull up **Socrative.com**  
and join room **XTEND159**

```
public void doQuickSort( int[] numbers, int start, int end ) {  
    If ( start < end ) {  
        int pivot = partition( numbers, start, end );  
        doQuickSort( start, pivot - 1 );  
        doQuickSort( pivot + 1, end );  
    } // end if  
}
```

# 159 XTENDED

## A WEEKLY REVIEW

### BINARY SEARCH

Pull up [Socrative.com](https://www.socrative.com)

and join room **XTEND159**

## ANOTHER CLASSIC COMPUTER SCIENCE PROBLEM...

- ▶ How can we take a list of numbers...

15, 8, 190, 252, 0, -9, 3001, 1, 99

and determine if a given number is in the list?

190 → true

290 → false

## ANOTHER CLASSIC COMPUTER SCIENCE PROBLEM...

- ▶ Option 1: Linear Search

15, 8, 190, 252, 0, -9, 3001, 1, 99

Walk through the list one number at a time.

190 → true

290 → false

THERE MUST BE A BETTER WAY...

A SKILLED 159 STUDENT

# INTRODUCING BINARY SEARCH

Pull up Socrative.com  
and join room **XTEND159**

```
private int binarySearch ( int[] numbers, int target, int left, int right ) {  
    int retVal;  
    int index = ( left + right ) / 2;  
  
    if ( left > right ) {  
        retVal = -1;  
    } else {  
        if (target == numbers[index] ) {  
            retVal = index;  
  
        } else if (target > numbers[index] ) {  
            retVal = recursiveBinarySearch (numbers, target, index + 1, right);  
  
        } else {  
            retVal = recursiveBinarySearch (numbers, target, left, index - 1);  
  
        }  
    }  
    return retVal;  
}
```

# BINARY SEARCH

- ▶ Binary Search Dance

# BINARY SEARCH

Pull up Socrative.com  
and join room **XTEND159**

- When searching for 23 in the following list using Binary Search:

2, 5, 8, 12, 16, 23, 38, 56, 72, 91

How many comparisons does it take to find the number **23**?