

## Sorting Practice

- Given the following list of integers

| 9 | 5 | 6 | 1 | 10 | 15 | 2 | 4 |

- Sort the list using **bubble sort**. Show your work!
- Sort the list using **insertion sort**. Show your work!
- Sort the list using **merge sort**. Show your work!
- Describe the best- and worst-case time and space complexity for each of these sorting algorithms as well as for **selection sort**.

## Induction Practice

- Prove that merge sort time complexity is  $O(n \log(n))$ 
  - Prove for  $n = 2^k$  (it is true for other  $n$ , but harder to prove)
  - That is, show that merge sort performs at most:
    - $n * \log(n) = 2^k * k$  comparisons of elements

## Recursion Practice

- Write a recursive method that duplicates the elements in a SLL *in place*.

```
public void doubleList(SLL<E> list) { ... }
```

- *Example:* Given an SLL of integers:

```
list = [1, 2, 3, 4]
```

calling `doubleList(list)` would modify `list` to be:

```
[1, 1, 2, 2, 3, 3, 4, 4]
```

Notes:

## More Recursion Practice

- Write a recursive method that multiplies two numbers, a and b, using only addition:

```
public int multiply(int a, int b);
```

- Write a recursive method that prints the digits of a number in reverse order.

```
public void reverseDigits(int num);
```

- Write a recursive method that replaces all instances of value a with value b in a Vector:

```
public static <E> void replace(Vector<E> v, E a, E b);
```

## Complexity Practice

- What is the running time of the following method:

```
public void reduce(int n) {  
    int result = 0;  
    while (n > 1) {  
        n = n / 2;  
        result = result + 1;  
    }  
    return result;  
}
```

## Complexity Practice

- True or false:

- $n^2 - 10n + 100$  is  $O(n^2)$

- $n$  is  $O(n \log_2(n))$

- $n^2$  is  $O(n^2 - 10n + 100)$

- $n \log_2(n)$  is  $O(n)$

- $\log_2(x)$  is  $O(x)$

- $x$  is  $O(\log_2(x))$

- $\sin(x)$  is  $O(1)$

‣ Note:  $f(x)$  is  $O(1)$  if  $f(x) \leq c$  for some constant  $c > 0$  and all large enough  $x$

Notes: