# CS 61BL Lab 5

Slides available at rpurp.com

Your partner today will be your project partner *and* your lab partner for this week.

Announcements

- Congratulations on surviving week 1!

- Quiz 1 scores have been released!

- Project 1 is (or will be soon) released! It's a partner project. Your project partner is also this week's lab partner.

MINIMUM	**MEDIAN**	**MAXIMUM**	**MEAN**	**STD DEV**
1.248	**2.33**	**3.0**	**2.37**	**0.41**
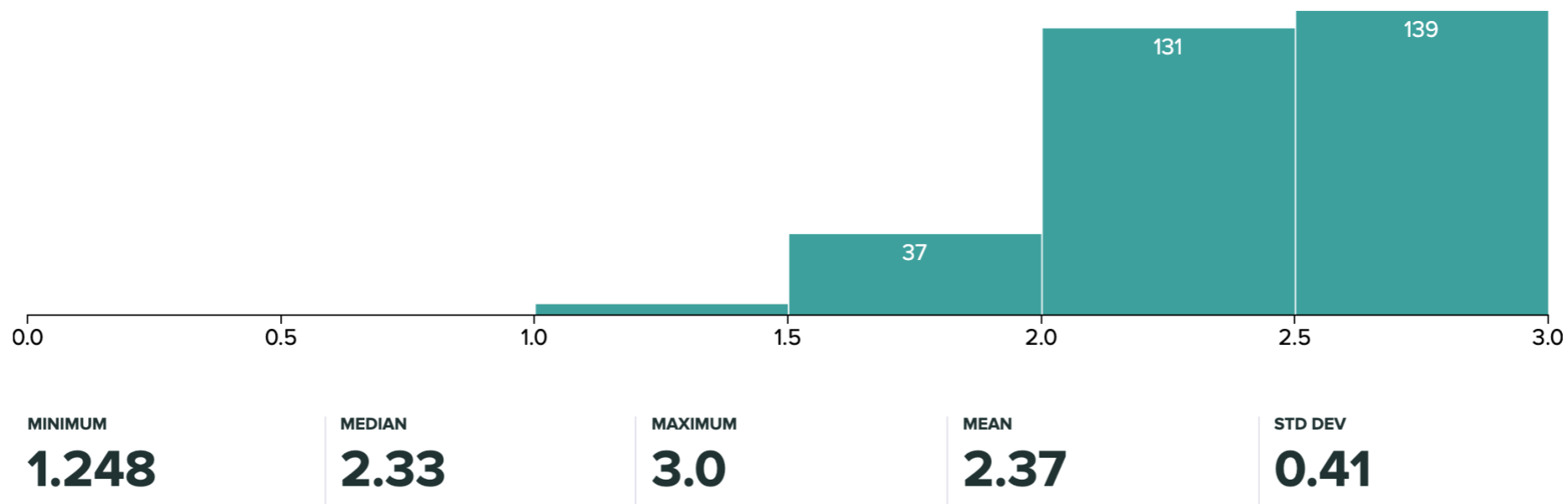
Linked Lists

- Linked Lists are a data structure that represents a list of items

- They are implemented using *nodes* that store an item and a reference to the next node

- Here is a minimal implementation of LinkedLists:

- **<u>Demo link</u>**

```java
public class IntList {
    public int item;
    public IntList next;
}
```

Linked List Demo

Creating Linked List Algorithms

- Oftentimes you can solve linked list problems iteratively or recursively

- Example: Linked List length

Iterative Linked List Size + Debugger Demo

Recursive Linked List Length

- First, an analogy: imagine I am in a very long line and want to know how many people are in line.

- Idea: Ask the person in front of you how many people there are in front of me.

- That person won't know either, so they'll ask the same how many people are in front of *them* to the next person.

- The person in front knows there is nobody in front of them, so can answer the previous person "There is one person in front of you"

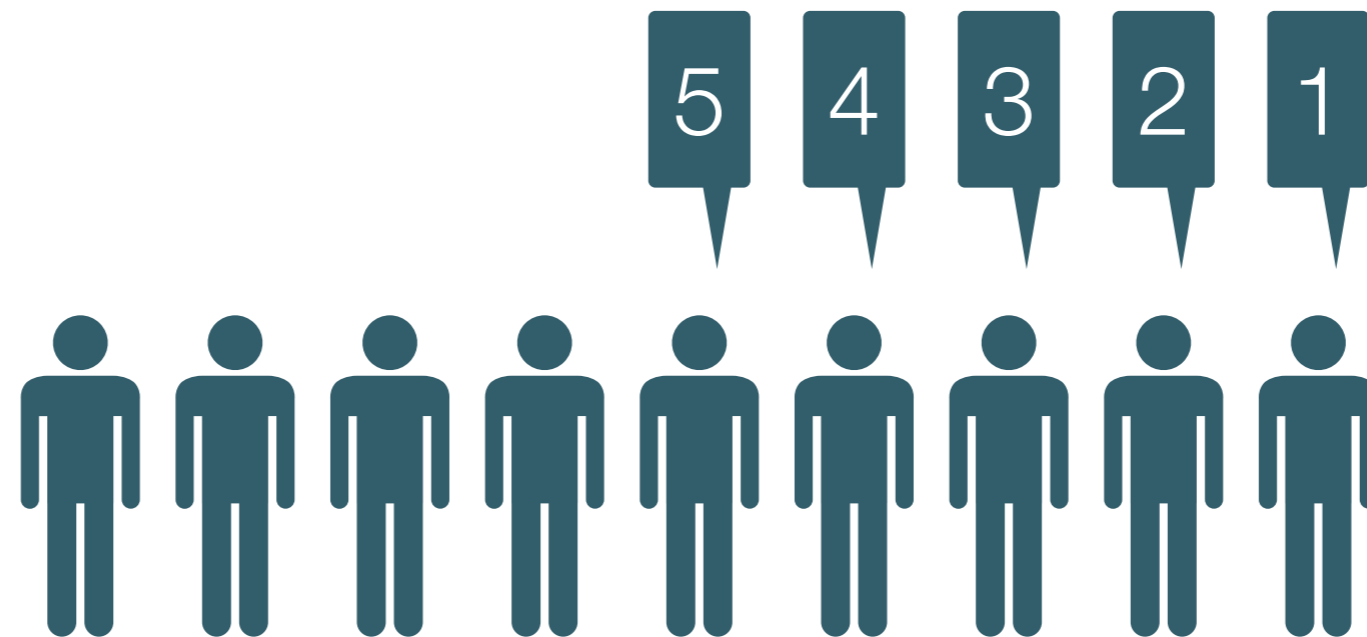# In Pictures

In Pictures

In Pictures

In Pictures

In Pictures

In Pictures

In Pictures

In Pictures

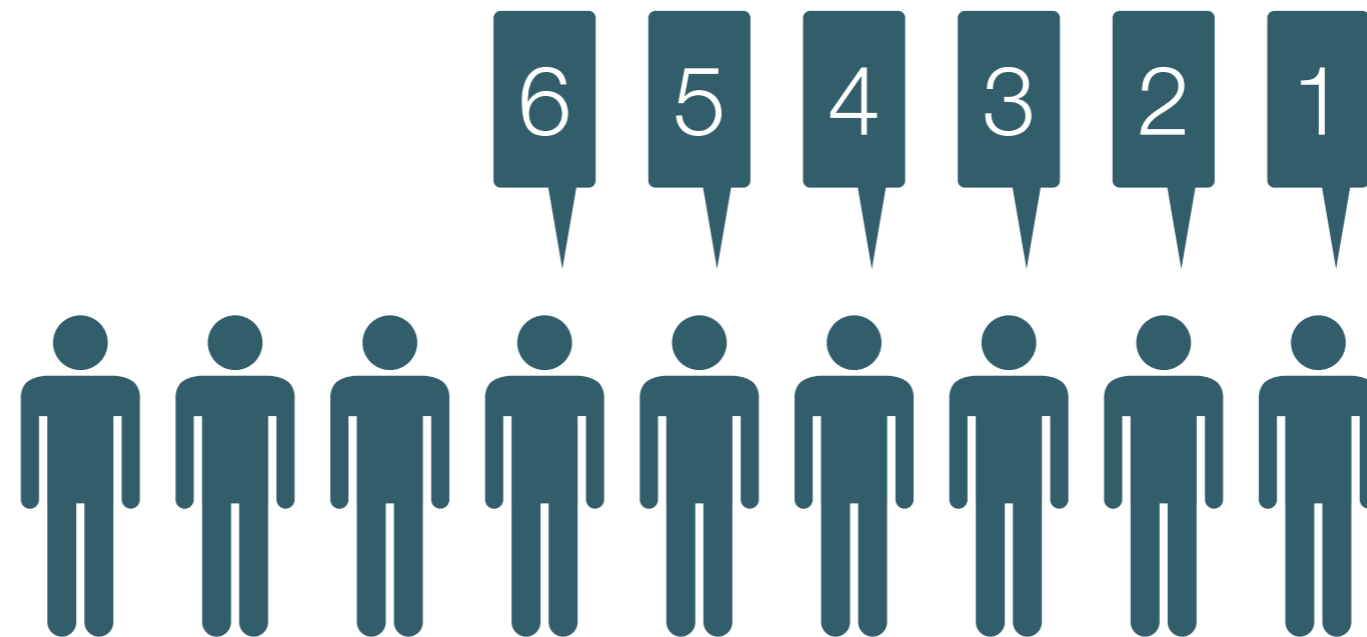# In Pictures

In Pictures
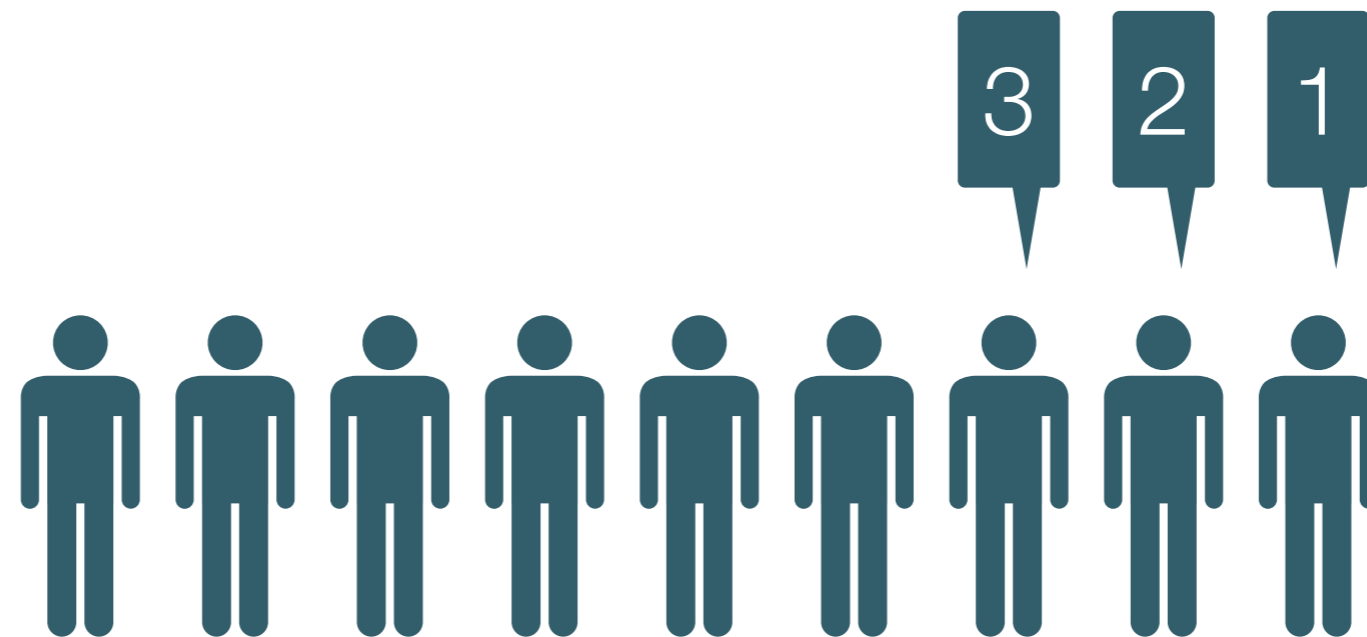
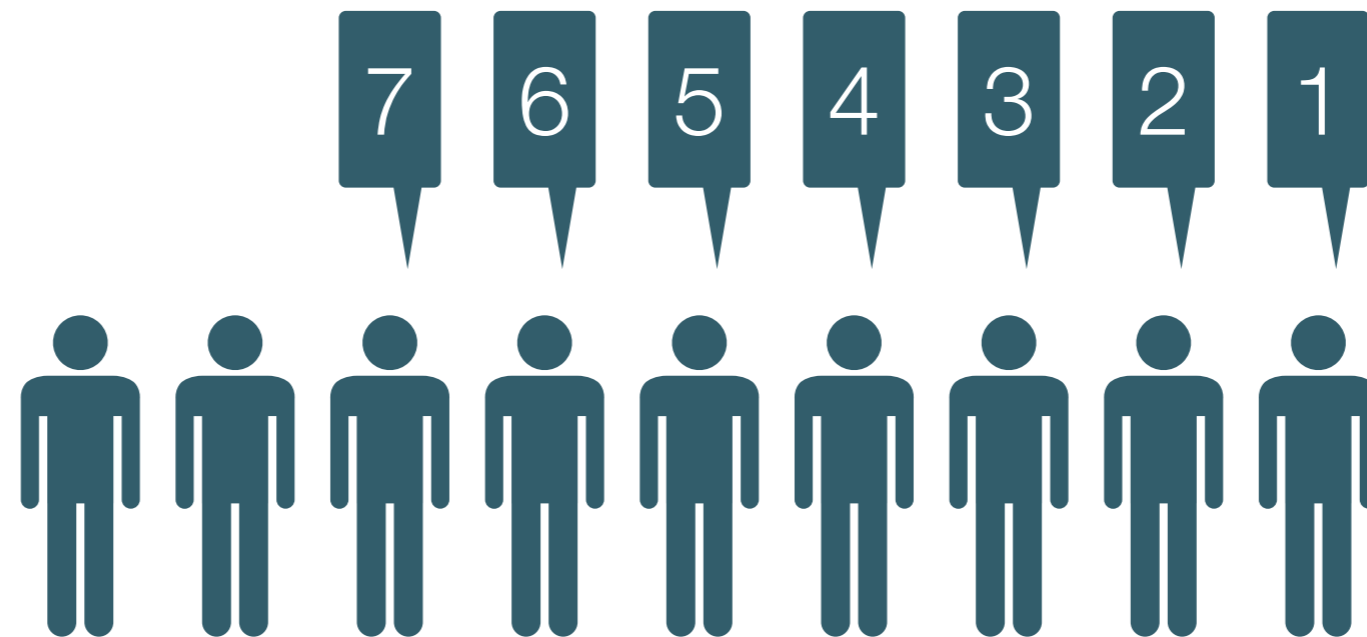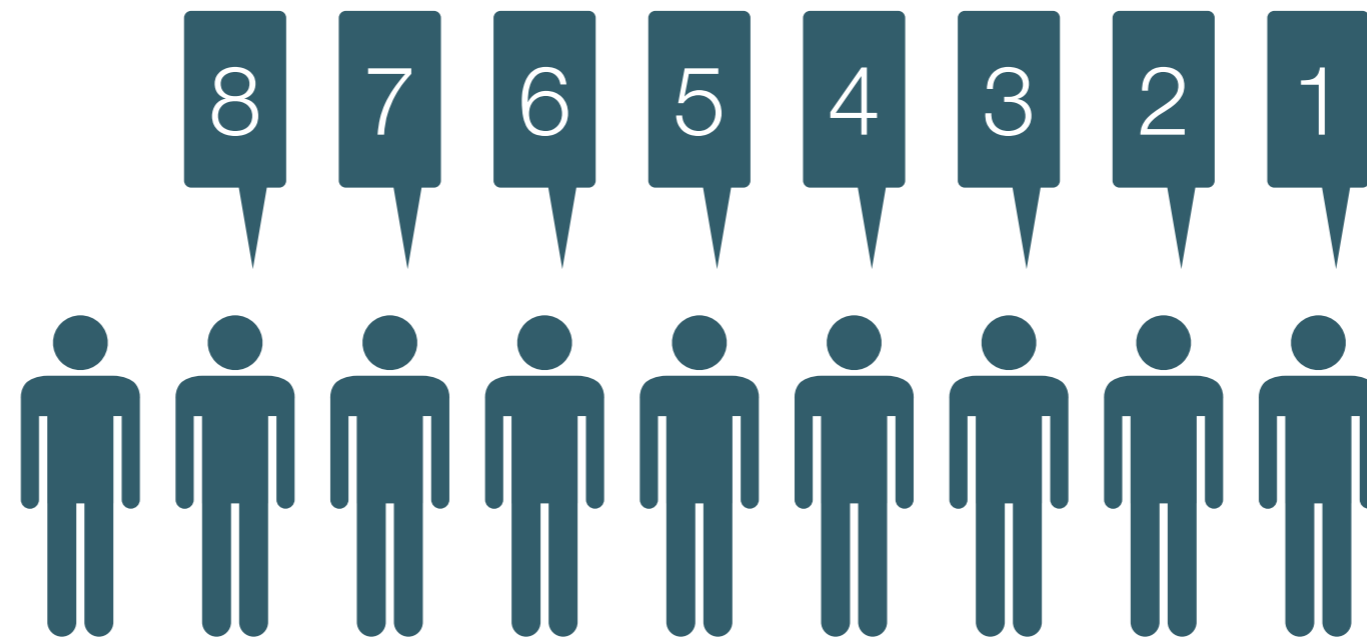# In Pictures
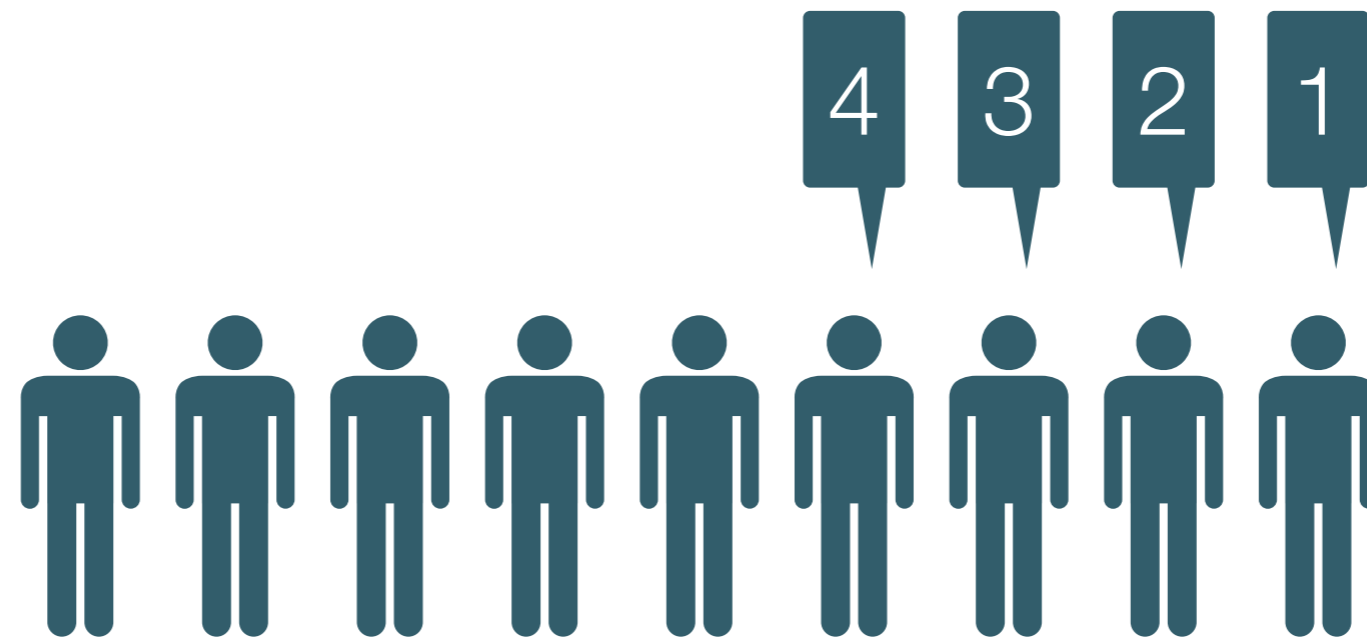
In Pictures

In Pictures

In Pictures

In Pictures

In Pictures

Recursive Linked List Size + Testing Demo