Searching

Reese Hatfield

Searching

- Ever used ArrayList.indexOf(Element)?
- Useful method when:
  - You need to check if the value is present
  - Where that value actually is

Searching

- IndexOf essentially just searches the list to find where the element is

- How could we implement this ourselves?

Searching

- Well you would just iterate thru it right?

- Often times, this is the best approach
- Let’s do it, but:
  - Let’s count the #operations
  - Operation = a check if cur == target

Searching

- This approach is often called a “Linear Search”

- If your list has N elements, it will take you time proportional to N to find your target
- O(N) time complexity

Searching

- Is this *really* the best way?
- If I gave you a dictionary, and said lookup the definition of “University”
- What would you do?

Searching

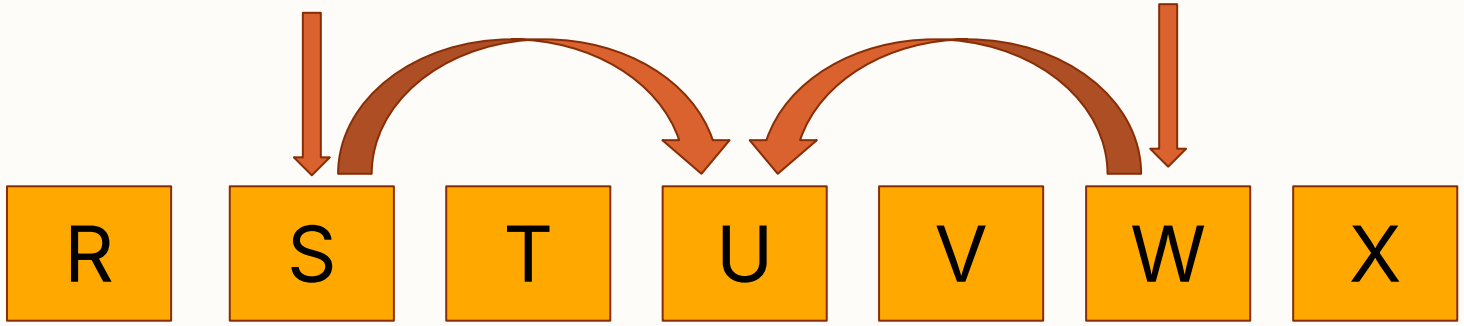
- What you're *not* going to do is start on page 1
  - Aardvark
  - Above
  - Abyss
  - ...
  - University

Searching

- That would take forever
- You’re probably gonna start somewhere in the middle
- Adjust according to wherever you happened to land

Searching

- If you landed at
  - S ⇒ go right
  - W ⇒ go left

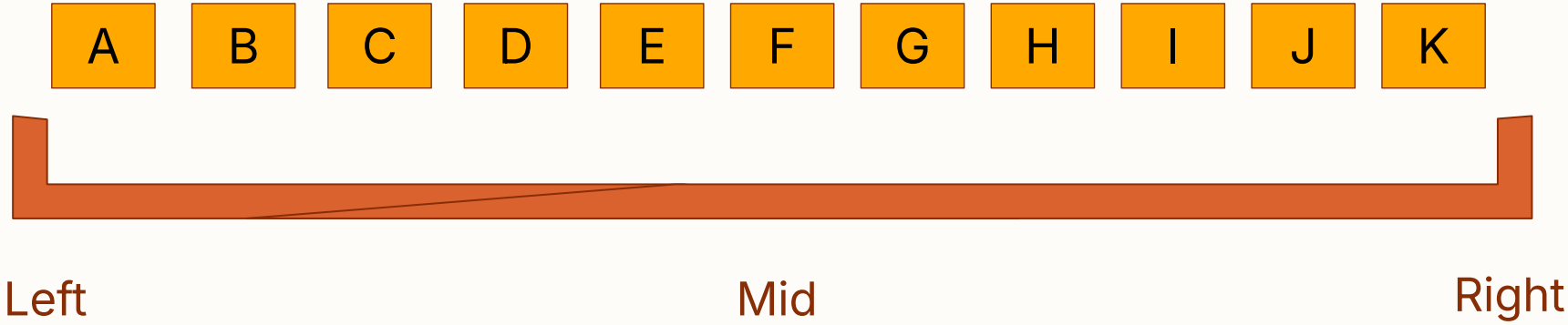


Searching

- We are only able to do this since we *know* the list is in alphabetical order
- To code something like this
  - We can narrow a sliding window

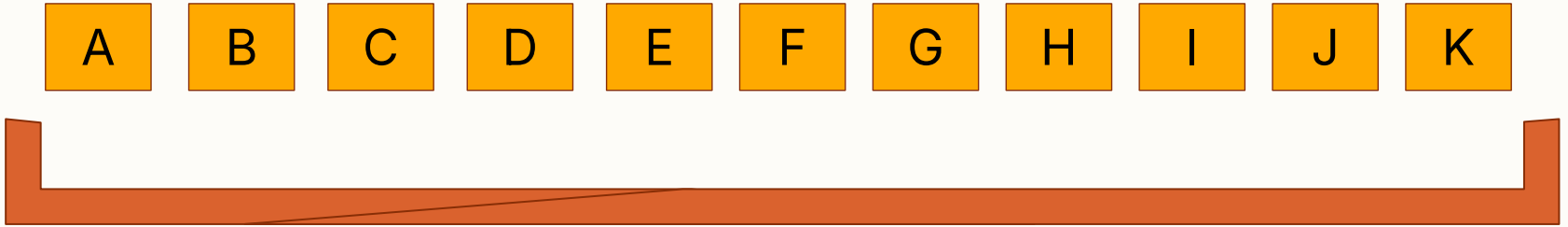
Searching

Target = J



Searching

Target = J



Left

Mid

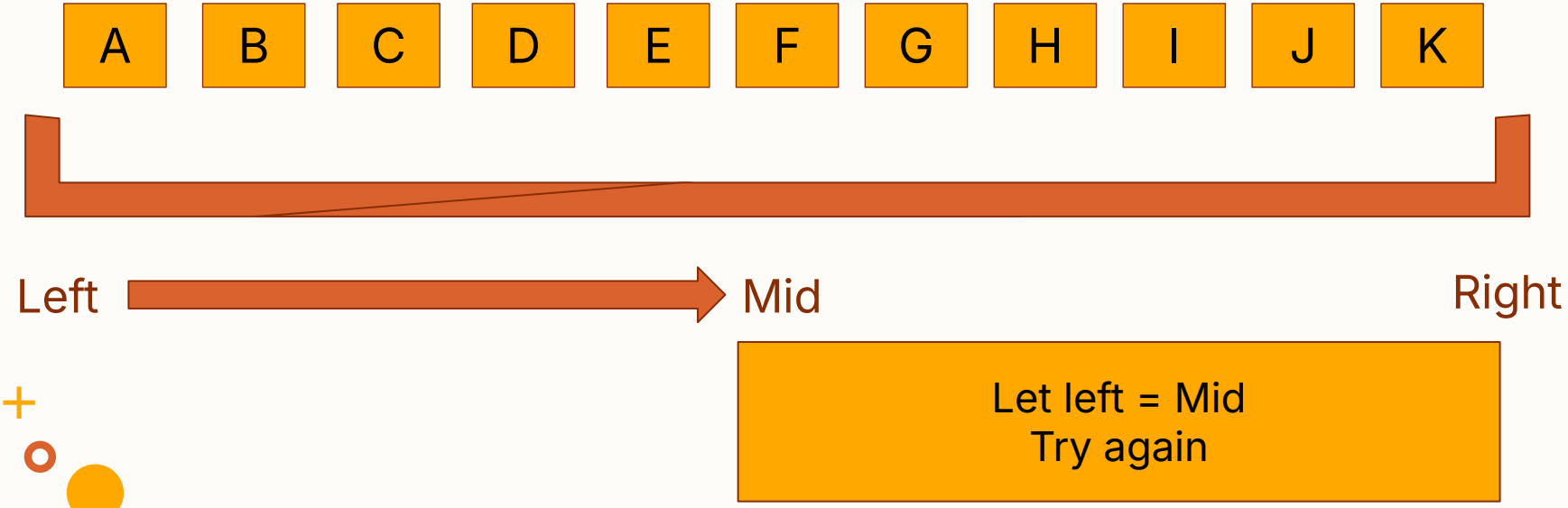
Right

“J” must be in this range

Since J > F

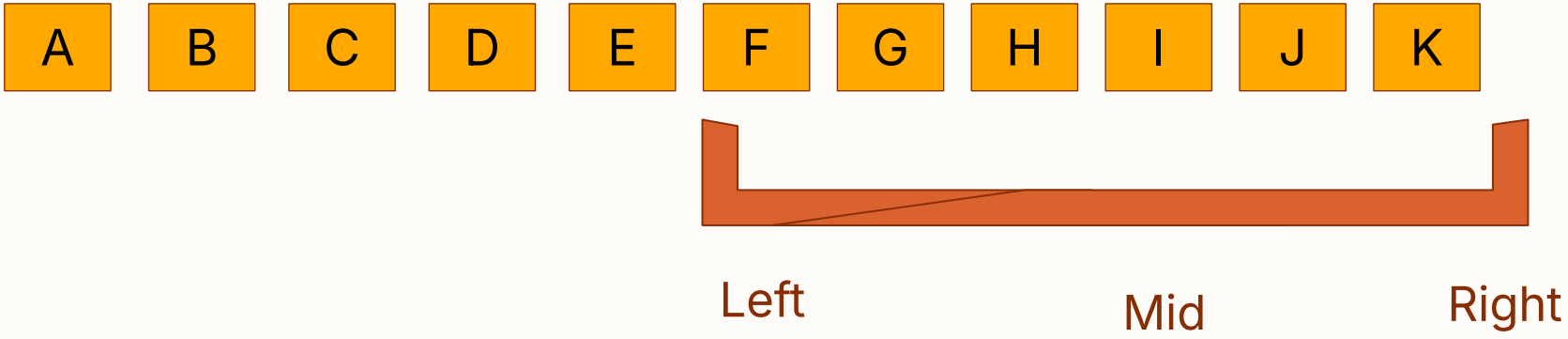
Searching

Target = J



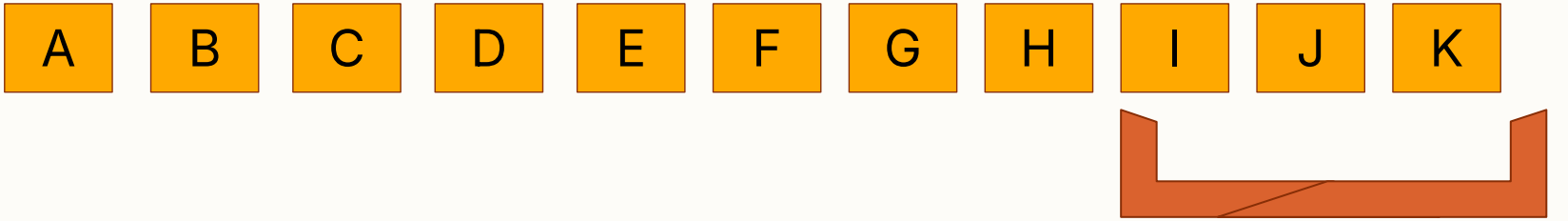
Searching

Target = J



Searching

Target = J



Eventually, Mid should equal the target, if it exists

Searching

- This type of search is called “Binary Search”
- Cut the space in half each time
  - Much more efficient
- Let’s code it
  - Count the operations
  - Inherently self similar

Searching

- Much faster than Linear Search
- If you have N elements, you need to do approximately log(N) operations
- O(logN)

Searching

- This comes up all the time
- Very commonly used
- Only possible is list is sorted

Afterwards I found a chatroom thread among Cambridge computer scientists, one of whom had also been told that unless he could pin down the moment of theft no one would look at the footage. He said he had tried to explain sorting algorithms to police — he was a computer scientist, after all.

You don't watch the whole thing, he said. You use a binary search. You fast forward to halfway, see if the bike is there and, if it is, zoom to three quarters of the way through. But if it wasn't there at the halfway mark, you rewind to a quarter of the way through. It's very quick. In fact, he had pointed out, if the CCTV footage stretched back to the dawn of humanity it would probably have only taken an hour to find the moment of theft. This argument didn't go down well.