How do we solve problems recursively?

Let's play this out a little further.

key < list[500,000]

Now let's look for key in the first half of the list. What index should we look at?
Why don't we look at the middle again, index 250,000?

key < list[250,000]

Keep going, but now only look at indices < 250,000.
Go for the middle, look at index 125,000

key > list[125,000]

We still didn't find key, but if key is in the list we know it comes after index 125,000, but before index 250,000. So we should search in the range [125,001, 249,999]

At this point, we've looked at 3 elements, list[500,000], list[250,000] and list[125,000].

But we've been able to remove 875,000 values from consideration! The total *search space* has gone from 1 million possible elements to 125,000 (just those elements between 125,000 and 250,000).

This is called the ***binary search*** algorithm because each time you look at a specific value, you split the list into 2 parts, and you get to throw out an entire half!

How do we solve problems recursively?