Binary Search

Search is the act of looking for a particular element in an array. This ends up being really similar to sorting, just instead of doing a whole array, we're just looking for one element in an array.

There are essentially two common ways of doing search: linear search and binary search. The former is the simplest code and really only useful if the list you're searching on is not sorted in any way. You just go through from 0 to the length of the array and ask "is the is the element I'm looking for?" No frills here. Its complexity is O(n).

Binary search is a bit more interesting. It only works if the array is already sorted. To explain it, let's take the example of how you find a name in a telephone book (if you even know what that is anymore!) A telephone book is a sorted list of names. You'll open the book more or less to the middle (or say you do, for argument's sake.) From there, if the name you're looking for is smaller/earlier in the alphabet, you'll go halfway to the smaller/earlier side of the book, and so-on-and-so-forth, keeping going halfway until eventually you land on the name you're looking for. Let's see how that works in practice.

```
[0, 5, 10, 12, 15, 19, 21, 22, 24, 30]

search for 12

start in the middle, is 19 === 12? no, smaller, go left

look in the middle of the smaller half, 10 === 12? no, larger, go right

look in the middle of the larger half (which is now just one number), is 12 ==
```