Binary Search

Input: array = [0, 1, 21, 33, 45, 45, 61, 72, 73]
target = 33

Output: 3

Input: A sorted array of integers and a target integer
Output: -1 if target integer is not in array or return the target integer index if it is in the array

Must use the Binary Search Algorithm

```
// O(logn) time | O(1) space
function binarySearch(array, target) {
  let left = 0;
  let right = array.length - 1;
  let mid;

  while (left <= right) {
    mid = Math.floor((left + right) / 2);
    if (array[mid] === target) {
      return mid;
    } else if (array[mid] < target) {
      left = mid + 1;
    } else if (array[mid] > target) {
      right = mid - 1;
    }
  }
  return -1;
}
```

Time: O(logn) since at each iteration, we cut the array into half

Space: O(1) since we do not use any more space as input size grows