App Academy

Binary Search

Binary Search

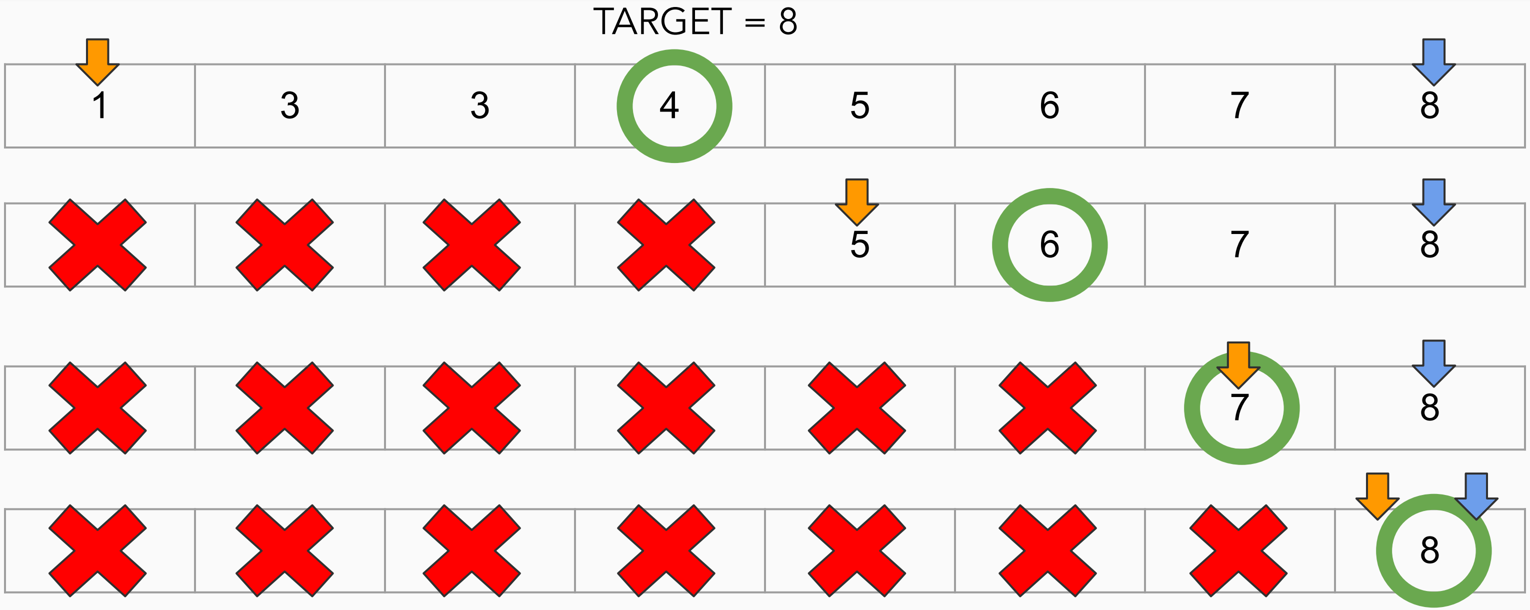
- Binary search is an extremely efficient searching algorithm.
- It is one of the few algorithms that can reach O(log(n)) runtime.
- The prerequisite for binary search is usually to have sorted array.
  - However there are some exceptions where you can binary search on things that aren't simply sorted arrays.

Binary Search

TARGET = 8

1	3	3	4	5	6	7	8

Binary Search example 1

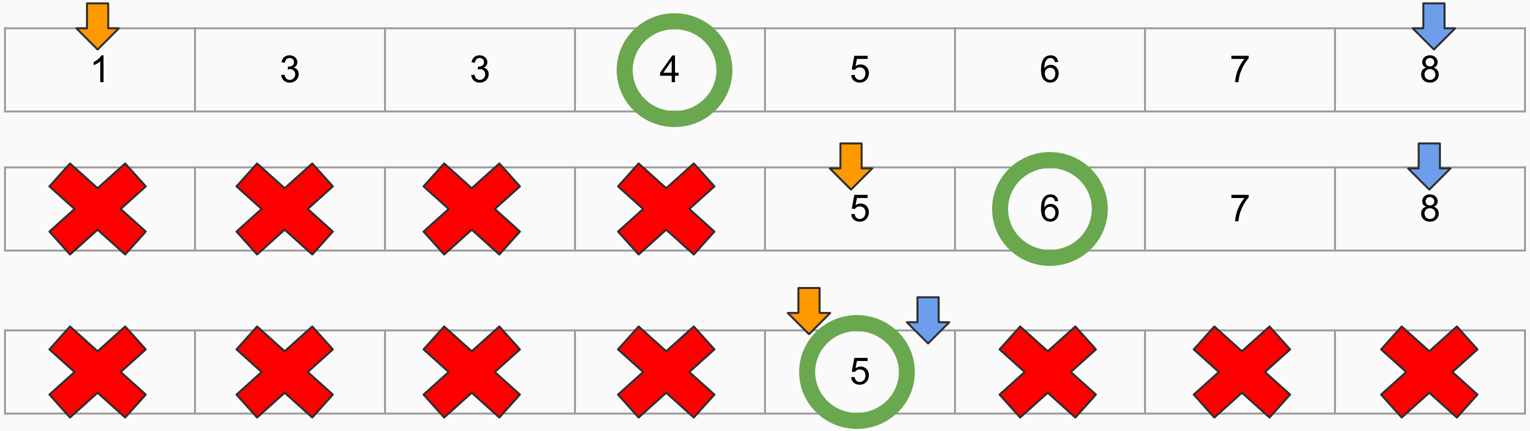


TARGET = 8

1	3	3	4	5	6	7	8
✗	✗	✗	✗	5	6	7	8
✗	✗	✗	✗	✗	✗	7	8
✗	✗	✗	✗	✗	✗	✗	8

Binary Search example 2

TARGET = 5



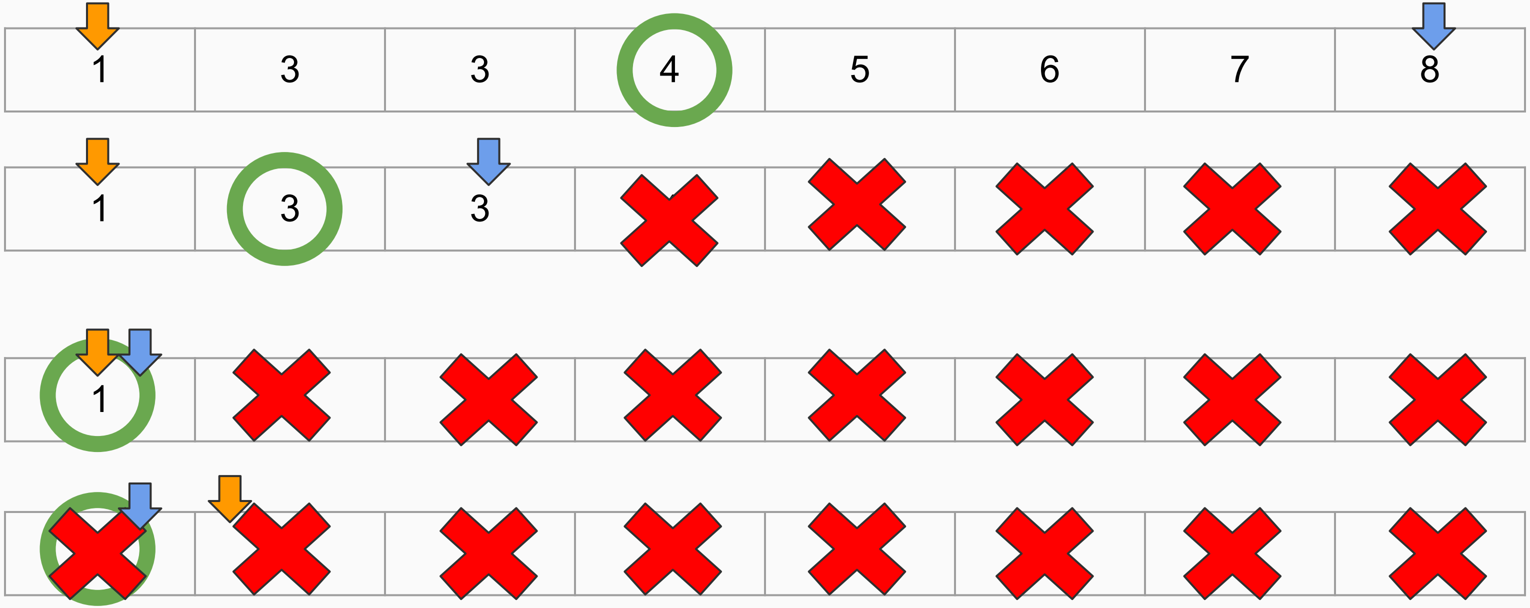
1	3	3	4	5	6	7	8

✖	✖	✖	✖	5	6	7	8

✖	✖	✖	✖	5	✖	✖	✖

Binary Search example 3

TARGET = 2



Demo

Binary Search

Recursive Binary Search

```
const search = function(nums, target, left = 0, right = nums.length) {
     if (left >= right) return nums[right] === target ? right : -1;
    const pivot = left + Math.floor((right - left)/2);
    if (nums[pivot] === target) return pivot;
    if (target < nums[pivot]) return search(nums, target, left, pivot - 1);
    if (target > nums[pivot]) return search(nums, target, pivot + 1, right);
 };
```

Iterative Binary Search

```
const search = function(nums, target) {
    let left = 0;
    let right = nums.length - 1;
    while (left <= right) {
        let pivot = left + Math.floor((right - left)/2)
        if (nums[pivot] === target) return pivot;
        if (target < nums[pivot]) right = pivot - 1;
        if (target > nums[pivot]) left = pivot + 1;
    }
    return -1;
};
```

Searching Ranges

Demo: Guess Number Higher or Lower

Questions?

Let's practice!

- Review
  - Search a 2D matrix
  - Find Peak Element
- Bonus
  - Koko Eating Bananas
  - Search in Rotated Sorted Array