

1 (https://www.interviewcake.com/question/javascript/find-rotation-point)                                   

I want to learn some big words so people think I'm smart.

I opened up a dictionary to a page in the middle and started flipping through, looking for words I didn't know. I put each word I didn't know at increasing indices in a huge array I created in memory. When I reached the end of the dictionary, I started from the beginning and did the same thing until I reached the page I started at.

Now I have an array of words that are mostly alphabetical, except they start somewhere in the middle of the alphabet, reach the end, and then start from the beginning of the alphabet. In other words, this is an alphabetically ordered array that has been "rotated." For example:

```
var words = [  
  'ptolemaic',  
  'retrograde',  
  'supplant',  
  'undulate',  
  'xenoepist',  
  'asymptote', // <-- rotates here!  
  'babka',  
  'banoffee',  
  'engender',  
  'karpatka',  
  'othellolagkage',  
];
```

▼ JavaScript

Write a function for finding the index of the "rotation point," which is where I started working from the beginning of the dictionary. This array is huge (there are lots of words I don't know) so we want to be efficient here.

Gotchas

We can get $O(\lg n)$ time.

Breakdown

The set is *mostly* ordered. We should exploit that fact.

What's a common algorithm that takes advantage of the fact that a set is sorted to find an item efficiently?

Binary search! We can write an adapted version of binary search for this.

In each iteration of our binary search, how do we know if the rotation point is to our left or to our right?

Try drawing out an example array!

```
words = [ 'k', 'v', 'a', 'b', 'c', 'd', 'e', 'g', 'i' ]
```

▼ JavaScript

If our "current guess" is the middle item, which is 'c' in this case, is the rotation point to the left or to the right? How do we know?

Notice that every item to the *right* of our rotation point is always alphabetically *before* the first item in the array.

So the rotation point is to our *left* if the current item is less than the first item. Else it's to our right.

Solution

This is a modified version of [binary search](#). At each iteration, we go right if the item we're looking at is greater than the first item and we go left if the item we're looking at is less than the first item.

We keep track of the lower and upper bounds on the rotation point, calling them `floorIndex` and `ceilingIndex` (initially we called them "floor" and "ceiling," but because we didn't imply the type in the name we got confused and created bugs). When `floorIndex` and `ceilingIndex` are directly next to each other, we know the floor is the last item we added before starting from the beginning of the dictionary, and the ceiling is the first item we added after.

```
function findRotationPoint(words) {  
  const firstWord = words[0];  
  
  var floorIndex = 0;  
  var ceilingIndex = words.length - 1;  
  
  while (floorIndex < ceilingIndex) {  
  
    // guess a point halfway between floor and ceiling  
    var guessIndex = Math.floor(floorIndex + ((ceilingIndex - floorIndex) / 2));  
  
    // if guess comes after first word  
    if (words[guessIndex] > firstWord) {  
      // go right  
      floorIndex = guessIndex;  
    } else {  
      // go left  
      ceilingIndex = guessIndex;  
    }  
  
    // if floor and ceiling have converged  
    if (floorIndex + 1 === ceilingIndex) {  
  
      // between floor and ceiling is where we flipped to the beginning  
      // so ceiling is alphabetically first  
      break;  
    }  
  }  
  
  return ceilingIndex;  
}
```

▼ JavaScript

Complexity

$O(\lg n)$ time and $O(1)$ space, just like binary search.

We're assuming that our word lengths are bound by some constant—if they were bounded by a non-constant l , each of our string comparisons would cost $O(l)$, for a total of $O(l * \lg n)$ runtime.

Type code  **Bonus**

This function *assumes* that the array is rotated. If it isn't, what index will it return? How can we fix our function to return 0 for an unrotated array?

Did you get it right?



Yes, I'm expert on this



Not quite, review later

Like this problem? Pass it on!



Share



Tweet

(https://www.facebook.com/sharer/sharer.php?u=https://www.interviewcake.com/question/javascript/rotation-point)

(https://www.facebook.com/sharer/sharer.php?u=https://www.interviewcake.com/question/javascript/rotation-point) rotation-point) rotation-point)

Yo, follow along!



(https://www.facebook.com/interviewcake) (https://www.facebook.com/interviewcake)

Subscribe to our weekly question email list » (/free-weekly-coding-interview-problem-newsletter)

Programming interview questions by company:

- Google interview questions (/google-interview-questions)
- Facebook interview questions (/facebook-interview-questions)
- Amazon interview questions (/amazon-interview-questions)

Programming interview questions by language:

- Java interview questions (/java-interview-questions)
- Python interview questions (/python-interview-questions)
- Ruby interview questions (/ruby-interview-questions)
- JavaScript interview questions (/javascript-interview-questions)
- **NEW:** Testing and QA interview questions (/testing-and-qa-interview-questions)
- **NEW:** SQL interview questions (/sql-interview-questions)

Copyright © 2016 Cake Labs, Inc. All rights reserved.
228 Park Ave S #82632, New York, NY US 10003 (804) 876-2253
Privacy (/privacy-policy) | Terms (/terms-and-conditions)