

#Programming

#JavaScript

#WebDev

#js

#algorithm

#algorithms

2 more ...

[Edit]

EN

# JavaScript - binary search algorithm example

1 contributors

2 contributions

0 discussions

Created by:

Aleena

454

★ 14

points



In this short article, we would like to show the main idea of the binary search algorithm providing example implementation in JavaScript.

## 1. Overview Edit

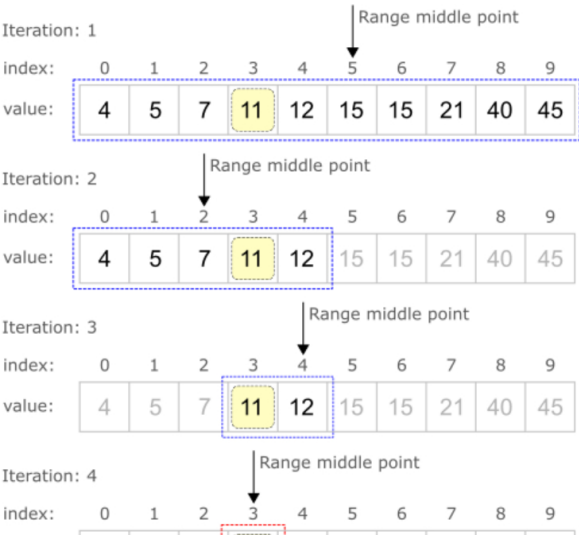
Binary search is a fast search algorithm with run-time complexity  $O(\log n)$ . This algorithm requires working on sorted data with random access to each element.

## 2. Description Edit

The main idea of the binary search algorithm is to:

- find middle point index in actual range with the formula:  
`var point = Math.ceil((index + limit) / 2);`  
Notes:
  - actual range has been marked with a **blue dashed border** on the below image,
  - middle point index has been marked with a **black arrow** on the below image,
  - in the beginning, the actual range contains all array elements ( `index=0` , `limit=array.length-1` ),
  - the middle point index is used to split the searching problem into 2 smaller ranges - only if it is necessary for the next steps,
- if the middle point value is equal to the expected value, the algorithm returns the middle point index,  
Notes:
  - expected value has been marked with a **yellow rounded rectangle** on the below image,
- if the middle point value is not equal to the expected value, the range is split into left or right ranges with an excluded middle point, a new range that is likely to contain the expected value is selected as an actual range and we go back to 1. point (we work all-time on sorted data).  
Notes:
  - if the expected value is smaller than the middle point value we select left range,
  - if the expected value is bigger than the middle point value we select right range,

Algorithm iterations:



Donate to Dirask

Our content is created by volunteers - like Wikipedia. If you think, the things we do are good, donate us. Thanks!

Donate



Direct link



SUBSCRIBE

value: 4 5 7 11 12 15 15 21 40 45

↑  
Middle point value equal  
to expected value



The binary search algorithm in JavaScript

SUBSCRIBE

Join to our subscribers to be up to date  
with content, news and offers.

Enter your e-mail

Subscribe

### 3. Custom iterative implementation example [Edit](#)

```
1 function searchIndex(array, value) {
2   var index = 0;
3   var limit = array.length - 1;
4   while (index <= limit) {
5     var point = Math.ceil((index + limit) / 2);
6     var entry = array[point];
7     if (value > entry) {
8       index = point + 1;
9       continue;
10    }
11    if (value < entry) {
12      limit = point - 1;
13      continue;
14    }
15    return point; // value == entry
16  }
17  return -1;
18 }
19
20
21 // Usage example:
22
23 // for this implementation array must be sorted from smallest to biggest!
24 var array = [ 4, 5, 7, 11, 12, 15, 15, 21, 40, 45 ];
25 var index = searchIndex(array, 11); // we want to find index for 11
26
27 console.log(index); // 3
```

Run

☐ Auto running

Reset

#### Jobs you may like



**Front-End Developer**  
Undisclosed salary  
Binance Remote



**Java Backend Engineer - Russia R...**  
Undisclosed salary  
Revolut Remote



**Java Engineer - Payments Integr...**  
Undisclosed salary  
Ocado Technology Barcelona, ES



**Frontend Developer(SE/SSE)**  
Undisclosed salary  
Paytm Bangalore, IN

[More job offers](#)

### Native Advertising



Get your tech brand or product in  
front of software developers.

For more information [Contact us](#)

### 4. Custom recursive implementation example [Edit](#)

```
1 function searchIndex(array, value) {
2   function split(index, limit) {
3     if (index > limit) {
4       return -1;
5     }
6     var point = Math.ceil((limit + index) / 2);
7     var entry = array[point];
8     if (value < entry) {
9       return split(index, point - 1);
10    }
11    if (value > entry) {
12      return split(point + 1, limit);
13    }
14    return point; // value == entry
15  }
16  return split(0, array.length - 1);
17 }
18
19
20 // Usage example:
21
22 // for this implementation array must be sorted from smallest to biggest!
23 var array = [ 4, 5, 7, 11, 12, 15, 15, 21, 40, 45 ];
24 var index = searchIndex(array, 11); // we want to find index for 11
25
26 console.log(index); // 3
```

Run

☐ Auto running

Reset

#### Posts you may like

- Python - check if file exists
- JavaScript - distance between points on the Earth in km
- TypeScript - subtract milliseconds from date
- Java stopwatch - how to measure elapsed time in java?
- JavaScript - math distance types formulas
- JavaScript - how to access this object inside event method?

[Show more](#)

**Dirask - we help you to  
solve coding problems.**

**Ask question.**



Join

## Features

[Home](#)  
[Snippets](#)  
[Wiki](#)  
[Questions](#)  
[Findings](#)

## About

[About & Contact](#)  
[Service Terms](#)  
[Cookie Policy](#)  
[Privacy Privacy](#)  
[GDPR Privacy Policy](#)  
[Policies](#)

## Other

[Code Formatters](#)  
[Funny Generators](#)  
[Post Lists](#)  
[Social Links](#)