Binary Search

Binary Search is one of the most popular searching algorithms. It is easy to grasp the knowledge on this algorithm intuitively. Binary Search is a divide-and-conquer algorithm, that divides the array roughly in half every time it checks whether an element of the array is the one, we're looking for. Binary search has a complexity of O(log n) in the worst case and O(1) in the best case.

Note: The input for a binary search algorithm should be a sorted list.

Below is an iterative approach to implement a binary search in JavaScript.

function binarySearch (sortedArray, key) {

    let high = sortedArray.length-1

    let low = 0;

    let str = "no match";

    while (low<=high) {

      let middle = Math.ceil((low+high)/2);

        if (sortedArray[middle] === key) {

            return middle;

        }

        else if (key < sortedArray[middle]) {

            high = middle-1;

        }

        else if (key > sortedArray[middle])

        {

            low = middle +1;

        }

    }

    return str;

    }

    let numbers = [1,2,3,4,7,9,12,15,35,67,89,100];

    let position = binarySearch(numbers,10);

    console.log(position);

The above implementation works for any array that is smaller but an array of size of the maximum possible integer value of 253 or largerwill make the calculation of middle position to become an array out of bound error.

i.e. middle = (low+high)/2

Therefore, it is always safe to use the following as a solution.

middle = low+ (high-low)/2