

**Федеральное государственное автономное образовательное учреждение  
высшего образования «Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»  
Институт радиоэлектроники и информационных технологий – РТФ**

**ЛАБОРАТОРНАЯ РАБОТА № 3  
Отчет**

**Решение алгоритмических задач на языке JavaScript**

Выполнил: Павленко Алексей Игоревич, РИМ-240950

Екатеринбург 2025

## Оглавление

Оглавление .....	2
Задачи .....	3
Ход работы.....	4
Вывод.....	6

## Задачи

1. Создать проект в WebStorm
2. Написать алгоритм сортировки массива. Оценить его с точки зрения времени выполнения и памяти
3. Реализовать функцию бинарного поиска в массиве
4. Функция проверки корректности постановки открывающихся и закрывающихся скобок в строке, где встречаются круглые, фигурные и квадратные скобки

## Ход работы

```
function bubbleSort(array, reversed) {
  for (let i = 0; i < array.length - 1; i++) {
    for (let j = 0; j < array.length - i - 1; j++) {
      if (typeof array[j] !== 'number') {
        throw new TypeError("Array should contain only numbers")
      }
      if (reversed && array[j] < array[j + 1] ||
        !reversed && array[j] > array[j + 1]) {
        [array[j], array[j + 1]] = [array[j + 1], array[j]];
      }
    }
  }
}
```

В качестве алгоритма сортировки была выбрана сортировка пузырьком. В данном случае сортируются исключительно числа, при нахождении какого-либо другого типа, будет поднято исключение. По-умолчанию выполняется сортировка по возрастанию, однако можно сортировать по убыванию передав в аргумент `reversed true`.

Временная сложность такого алгоритма квадратичная —  $O(n^2)$ . Если оценивать его с точки зрения памяти, то получится константная величина, поскольку используется только фиксированное количество числовых переменных.

```
function binarySearch(array, item, start = 0, end = array.length - 1) {
  if (start > end) {
    return -1
  }

  const middle = Math.floor((start + end) / 2)

  if (typeof array[middle] !== "number") {
    throw new TypeError("Array should contain only numbers")
  }
  if (array[middle] === item) {
    return middle
  }
  if (array[middle] < item) {
    return binarySearch(array, item, middle + 1, end)
  } else {
    return binarySearch(array, item, start, middle - 1)
  }
}
```

Алгоритм бинарного поиска так же написан только для чисел, и поднимет ошибку в случае нахождения не числа. Он также должен запускаться только на отсортированных по возрастанию массивах, внутри функции это не проверяется. Данная реализация алгоритма рекурсивна. При нахождении искомого элемента, будет возвращен его индекс, иначе -1.

```
function bracketCheck(str) {  
  const bracketsStack = []  
  const pairs = {  
    ')': '(',  
    '}': '{',  
    ']': '['  
  };  
  for (const c of str) {  
    if (['(', '{', '['].includes(c)){  
      bracketsStack.push(c)  
      continue  
    }  
    if ([')', '}', ']').includes(c) && bracketsStack.pop() !== pairs[c]){  
      return false  
    }  
  }  
  return true  
}
```

Поиск пар скобок в строке производится проходом по всей строке. Найденные открывающие скобки вносятся в стек. В качестве стека используется array, поскольку он реализует необходимые методы (push и pop). При нахождении закрывающей скобки, производится сравнение с последней открывающей скобкой из стека с помощью словаря. Если типы скобок совпадают, то алгоритм продолжает работу, иначе возвращается false. Любые другие символы игнорируются.

## **Вывод**

В данной лабораторной работе, были реализованы алгоритмы сортировки пузырьком, бинарного поиска и проверки корректности скобок в строке. Для сортировки были приведены сложность по времени и памяти.

Ссылка на репозиторий: <https://github.com/AlexFaernon/JS-Lab-3>