# JavaScript and Data Structure

## Data Structure

### Problems

* 1. Is Unique: Implement an algorithm if a string has all unique character. What if you cannot use additional data structure?

|  |
| --- |
| const checkUniqueCharacters =(text) => {  //checking ASCII character  if(text.length > 128) {  return false;  }  let hash = {}  for(item of text)  {  //If the key is already present in the hash table return false  if(hash[text[item]]){  return false  }  else {  // storing the value in hashtable  /\*hash {  text[i]: i;  }\*/  hash[text[item]] = item;  }  }  return true;  }  console.log(checkUniqueCharacters('1232'));  //Ask is it ASCII or Unicode String, assume as it is ASCII where max character can be 128  //Use Hash table, hashing also leads to collision.  //Time Complexity is O(n) , but you can say it will O(c) where c is the size of the character set (128)  //Space complexity O(1)  // Also we can improve the Time Complexity to O(n log(n)), if we sort the text, but remember it sorting can take more space. |

* 1. Check Permutation: Given two strings, write a method to decide if one is permutation of the other?

|  |
| --- |
| const checkIsPermutationofString = (str1,str2) => {  if(str1.length !== str2.length) {  return false  }  else {  if(str1 === str2) {  return true  }  return Sort(str2) === Sort(str1);  }  }  const Sort = (str) => {  const array1 = str.split('');  const sortValue = array1.sort();  return sortValue.toString();  }  // Check the length of both the string is they are not matched then it is not permutated  // Sol 1:- Sort both the string and compare    console.log(checkIsPermutationofString('abe','cab')); |

## JavaScript

1. Covert string to array?

Using split()

const array1 = str.split('');