Report on All Functions in "Operations on Arrays"

This report provides an overview of all the functions under the "Operations on Arrays" section.

They are categorized into Basic, Intermediate, and Advanced operations.

---

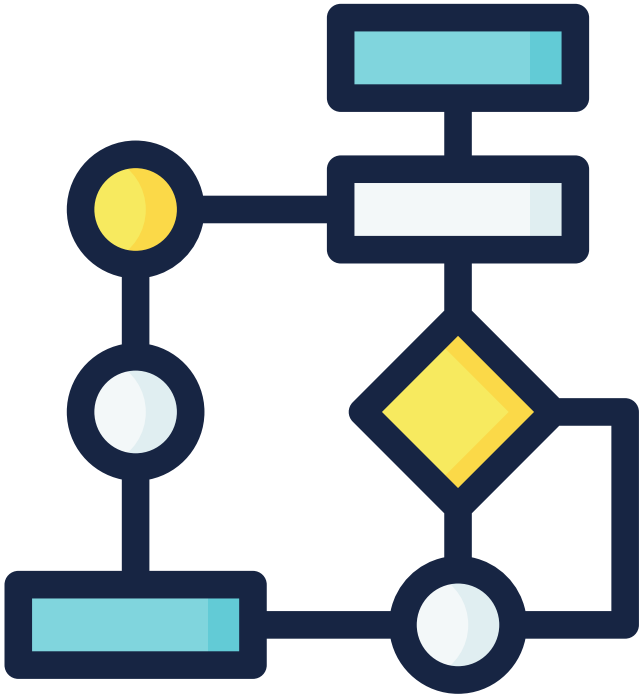
1. Basic Array Operations

1.1. Initialize Array

Signature: void initializeArray(int arr[], int size, int value)

Description: Initializes all elements of the array with a specified value.

Example: For size = 5 and value = 2, array = [2, 2, 2, 2, 2].



1.2. Print Array

Signature: void printArray(int arr[], int size)

Description: Prints all elements of the array.

1.3. Find Maximum

Signature: int findMax(int arr[], int size)

Description: Finds the maximum element in the array.

Example: In [1, 5, 3], the maximum is 5.

1.4. Find Minimum

Signature: int findMin(int arr[], int size)

Description: Finds the minimum element in the array.

Example: In [1, 5, 3], the minimum is 1.

1.5. Calculate Sum

Signature: int sumArray(int arr[], int size)

Description: Computes the sum of all elements in the array.

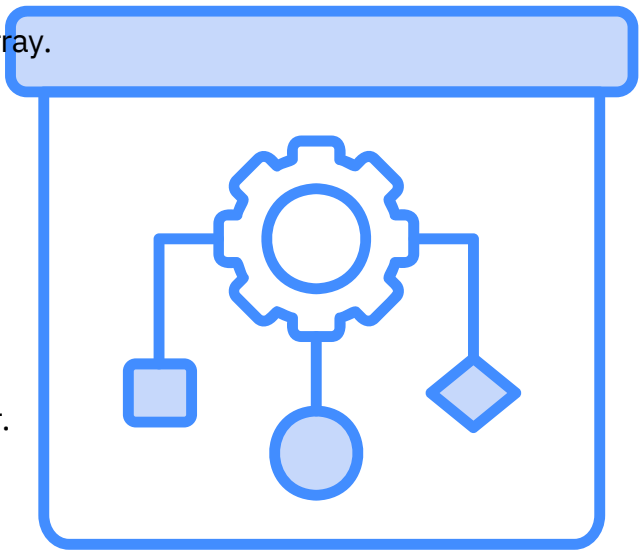
Example: For [1, 2, 3], the sum is 6.

1.6. Calculate Average

Signature: double averageArray(int arr[], int size)

Description: Computes the average of all elements in the array.

Example: For [1, 2, 3], the average is 2.



1.7. Check if Array is Sorted

Signature: bool isSorted(int arr[], int size)

Description: Checks if the array is sorted in ascending order.

Example: [1, 2, 3] is sorted, but [3, 2, 1] is not.

---

2. Intermediate Array Operations

2.1. Reverse Array

Signature: void reverseArray(int arr[], int size)

Description: Reverses the elements of the array.

Example: [1, 2, 3] becomes [3, 2, 1].

2.2. Count Even and Odd Numbers

Signature: void countEvenOdd(int arr[], int size, int* evenCount, int* oddCount)

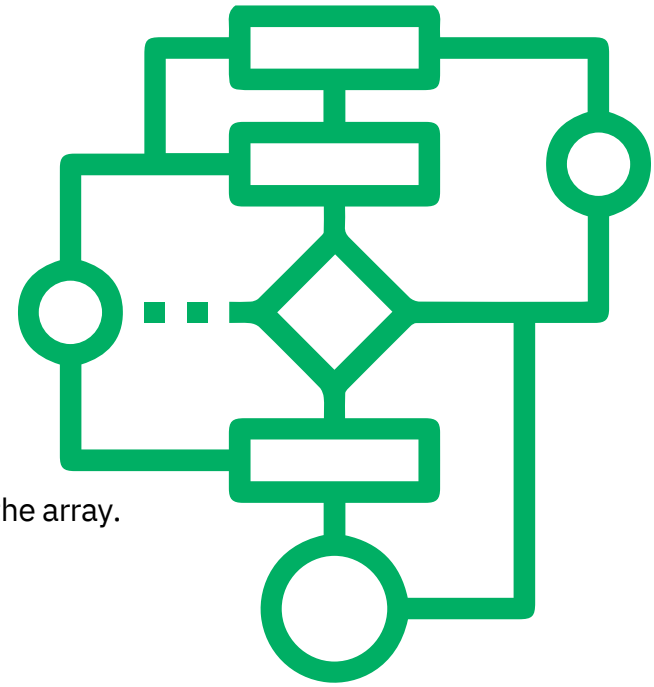
Description: Counts the number of even and odd elements in the array.

2.3. Find Second Largest Element

Signature: int secondLargest(int arr[], int size)

Description: Finds the second largest element in the array.

Example: In [1, 3, 2], the second largest is 2.



2.4. Find Frequency of Elements

Signature: void elementFrequency(int arr[], int size)

Description: Finds the frequency of each unique element in the array.

2.5. Remove Duplicates

Signature: int removeDuplicates(int arr[], int size)

Description: Removes duplicate elements from the array and returns the new size.

Example: [1, 2, 2, 3] becomes [1, 2, 3].

2.6. Binary Search

Signature: int binarySearch(int arr[], int size, int target)

Description: Performs binary search on a sorted array to find the target element.

2.7. Linear Search

Signature: int linearSearch(int arr[], int size, int target)

Description: Searches for a target element in the array using linear search.

2.8. Left Shift Array

Signature: void leftShift(int arr[], int size, int rotations)

Description: Shifts the array to the left by a specified number of positions.

Example: [1, 2, 3, 4] shifted left by 1 becomes [2, 3, 4, 1].

2.9. Right Shift Array

Signature: void rightShift(int arr[], int size, int rotations)

Description: Shifts the array to the right by a specified number of positions.

Example: [1, 2, 3, 4] shifted right by 1 becomes [4, 1, 2, 3].

---

3. Advanced Array Operations

3.1. Find Missing Number

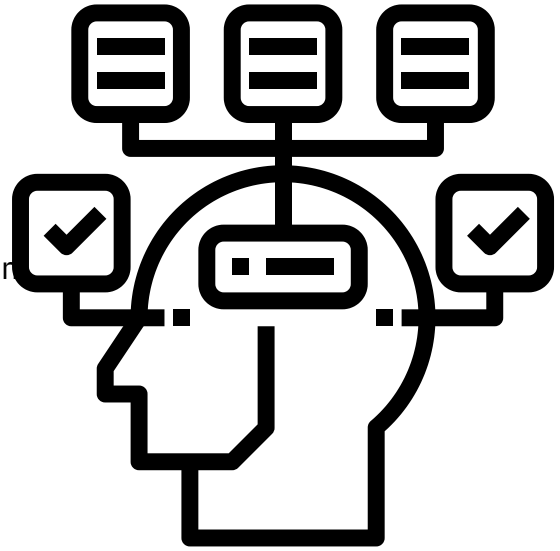
Signature: int findMissingNumber(int arr[], int size)

Description: Finds the missing number in an array of size n-1 containing numbers from 1 to n.

3.2. Find Pairs with Given Sum

Signature: void findPairsWithSum(int arr[], int size, int sum)

Description: Finds all pairs of elements whose sum equals the specified value.



3.3. Subarray with Given Sum

Signature: void findSubArrayWithSum(int arr[], int size, int sum)

Description: Finds a continuous subarray that adds up to the given sum.

3.4. Rearrange Positive and Negative Numbers

Signature: void rearrangeAlternatePositiveNegative(int arr[], int size)

Description: Rearranges the array such that positive and negative numbers alternate.

3.5. Find Majority Element

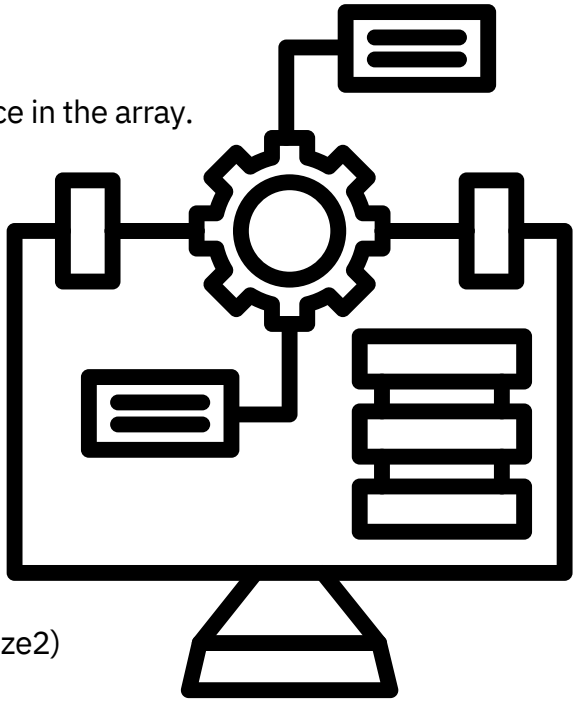
Signature: int findMajorityElement(int arr[], int size)

Description: Finds the majority element in the array (an element that appears more than n/2 times).

3.6. Longest Increasing Subsequence

Signature: int longestIncreasingSubsequence(int arr[], int size)

Description: Finds the length of the longest increasing subsequence in the array.



3.7. Find Duplicates

Signature: void findDuplicates(int arr[], int size)

Description: Identifies duplicate elements in the array.

3.8. Find Intersection of Two Arrays

Signature: void findIntersection(int arr1[], int size1, int arr2[], int size2)

Description: Finds the common elements between two arrays.

3.9. Find Union of Two Arrays

Signature: void findUnion(int arr1[], int size1, int arr2[], int size2)

Description: Finds the union of two arrays.

These functions provide a comprehensive toolkit for array manipulations and algorithms.

They are essential for solving various computational problems involving one-dimensional arrays.