Documentation

The problem "Contains Duplicate II" requires checking if a given integer array contains two distinct indices i and j such that the values at these indices are the same, and the absolute difference between i and j is less than or equal to a given integer k. In other words, the task is to determine if duplicate values in the array are positioned close enough to each other based on the provided k value.

To solve this problem efficiently, we can use a dictionary (or hashmap) to keep track of the last occurrence (index) of each element in the array. As we iterate through the array, we can check whether the current element has been seen before. If it has, we calculate the difference between the current index and the index where this element was last seen. If this difference is less than or equal to k, we immediately return True, indicating that there is a pair of duplicates whose indices meet the required condition. If the difference exceeds k, we update the dictionary to store the current index as the new last occurrence of the element. If the component has not been seen before, we simply add it to the dictionary and its index.

This approach ensures that we traverse the array only once, and dictionary operations (insertion and lookup) are performed in constant time on average, making the solution efficient with time complexity of O(n), where n is the length of the array. The space complexity is also O(n), as the dictionary stores each element and its corresponding index.

The algorithm proceeds as follows:

- First, we initialize an empty dictionary to keep track of the last-seen index of each element.
- We then iterate over the array, checking for each element whether it has already been encountered.
- If an element has been encountered before, we calculate the distance between its current index and the index where it was last seen. If this distance is less than or equal to k, we return True, meaning we have found two indices that satisfy the condition.
- If the distance exceeds k, we update the last seen index for that element in the dictionary.
- If no such pair of indices is found by the end of the loop, we return False.

This approach efficiently solves the problem within the given constraints, where the array length can be up to 100,000, and the element values can range from -10^9 to 10^9. The key insight is to use a hashmap to remember the index of each element and only focus on finding duplicates that are within a distance of k, allowing us to skip unnecessary comparisons and achieve the solution in linear time.