Pairs With Difference K

solved by	Senan
Platform	GeeksForGeeks
difficulty	Easy
tags	Hash Map
language	C++
solved on	@30/10/2024
link	https://www.geeksforgeeks.org/problems/pairs-with-difference-k1713/1
Completion	☑

Intuition

To count pairs with a specific difference **k** in an array, we can leverage a hash map to store the frequency of each element as we iterate through the array. This allows us to check for complementary pairs (elements that differ by **k)** in constant time, significantly improving efficiency over a brute-force approach.

Approach

1. Use an unordered map `mpp` to keep track of the frequency of each element as we iterate.
2. For each element `arr[i]`:
    - Check if there exists an element `arr[i] + k` in `mpp`. If so, increment `count` by the frequency of `arr[i] + k`.
    - Similarly, check if there exists an element `arr[i] - k` in `mpp` and increment `count` by its frequency.
3. Update the map with the current element `arr[i]` by increasing its frequency.
4. The total `count` will then represent the number of pairs with a difference of **k**.

Complexity

Time Complexity:

The time complexity is **O(n)**, where **n** is the size of the array, since we only perform constant-time operations for each element in the array.

Space Complexity:

The space complexity is **O(n)** for storing up to **n** elements in the hash map.

Code

```
class Solution {
  public:
    int countPairsWithDiffK(vector<int>& arr, int k) {
        int count = 0;
        unordered_map<int, int> mpp;
        for (int i = 0; i < arr.size(); i++) {
            if (mpp.count(arr[i] + k)) count += mpp[arr[i] + k];
```

```
            if (mpp.count(arr[i] - k)) count += mpp[arr[i] - k];
            mpp[arr[i]]++;
        }
        return count;
    }
};
```