Triplet Family

solved by	Senan
Platform	GeeksForGeeks
difficulty	Easy
tags	Logic
language	C++
solved on	@27/10/2024
link	https://www.geeksforgeeks.org/problems/triplet-family/1
Completion	☑

Intuition

The problem is to find if there exists a triplet (a, b, c) in the array such that $c = a + b$. This is essentially checking if the sum of any two elements in the array is present as another element in the array. Using a hash set for quick lookups allows us to efficiently check if any sum exists in the array.

Approach

1. Create an unordered set `mySet` and initialize it with all elements from `arr`. This will allow **O(1)** time complexity for checking if a specific sum is in the array.
2. Iterate over all pairs of elements in `arr` using two nested loops. For each pair (i, j):
    - Compute the sum of `arr[i]` and `arr[j]`.
    - Check if this sum exists in `mySet`. If it does, return `true`, as we have found a triplet that meets the condition $c = a + b$.
3. If no such triplet is found after checking all pairs, return `false`.

Complexity

Time Complexity:

The time complexity is **O(n^2)** for iterating over all pairs in the array. Hash set operations are **O(1)**, so checking the existence of the sum takes constant time for each pair.

Space Complexity:

The space complexity is **O(n)** due to the additional space used by `mySet` to store the elements of `arr`.

Code

```
class Solution {
public:
    bool findTriplet(vector<int>& arr) {
        unordered_set<int> mySet(arr.begin(), arr.end());
        for (int i = 0; i < arr.size(); i++) {
            for (int j = i + 1; j < arr.size(); j++) {
                if (mySet.find(arr[i] + arr[j]) != mySet.end()) {
                    return true;
                }
            }
        }
        return false;
    }
};
```