

2183. Count Array Pairs Divisible by K

[Leetcode Link](#)

Problem Description

Given an array `nums` and an integer `k`, we have to find the number of pairs `(i, j)` such that the product of `nums[i]` and `nums[j]` is divisible by `k`. Note that `k` can be as large as `1e5`.

Here's a high-level approach to solving this problem:

1. Calculate the greatest common divisor (gcd) of each value in `nums` with `k`.
2. Then, for each pair `(nums[i], nums[j])`, check if their gcd values multiplied together are divisible by `k`. If yes, add them to the count.

Let's walk through an example to better understand the approach.

Example

Suppose `nums = [1, 2, 3, 4, 5]` and `k = 6`. We first find the gcd values for each element in the array `nums` with `k`.

$1 \times 6 = 6$, $\text{gcd}(1, 6) = 1$ $2 \times 3 = 6$, $\text{gcd}(2, 6) = 2$ $3 \times 2 = 6$, $\text{gcd}(3, 6) = 3$ $4 \times 1.5 = 6$, $\text{gcd}(4, 6) = 2$ $5 \times 1.2 = 6$, $\text{gcd}(5, 6) = 1$

Now the gcd values are `[1, 2, 3, 2, 1]`.

We go through each pair of gcd values and check if their product is divisible by `k`. In our example:

`(1, 2)` - No `(1, 3)` - No `(1, 2)` - No `(1, 1)` - No `(2, 3)` - Yes `(2, 2)` - No `(2, 1)` - No `(3, 2)` - Yes `(3, 1)` - No `(2, 1)` - No

We find two pairs that are divisible by `k`: `(2, 3)` and `(3, 2)`. So the final answer is `2`.

Solution

Python

```
1
2 python
3 from math import gcd
4 from typing import List
5
6
7 class Solution:
8     def countPairs(self, nums: List[int], k: int) -> int:
9         ans = 0
10        gcds = {}
11
12        for num in nums:
13            gcd_num = gcd(num, k)
14            for gcd_val, count in gcds.items():
15                if gcd_num * gcd_val % k == 0:
16                    ans += count
17            gcds[gcd_num] = gcds.get(gcd_num, 0) + 1
18
19        return ans
```

Java

```
1
2 java
3 import java.util.HashMap;
4 import java.util.Map;
5
6 class Solution {
7     public long countPairs(int[] nums, int k) {
8         long ans = 0;
9         Map<Integer, Integer> gcds = new HashMap<>();
10
11        for (int num : nums) {
12            int gcdNum = gcd(num, k);
13            for (Map.Entry<Integer, Integer> entry : gcds.entrySet()) {
14                if (gcdNum * entry.getKey() % k == 0) {
15                    ans += entry.getValue();
16                }
17            }
18            gcds.put(gcdNum, gcds.getOrDefault(gcdNum, 0) + 1);
19        }
20
21        return ans;
22    }
23
24    private int gcd(int a, int b) {
25        if (b == 0) {
26            return a;
27        } else {
28            return gcd(b, a % b);
29        }
30    }
31 }
```

JavaScript

```
1
2 javascript
3 class Solution {
4     countPairs(nums, k) {
5         let ans = 0;
6         const gcds = new Map();
7
8         for (const num of nums) {
9             const gcdNum = this.gcd(num, k);
10            for (const [gcdVal, count] of gcds.entries()) {
11                if (gcdNum * gcdVal % k === 0) {
12                    ans += count;
13                }
14            }
15            gcds.set(gcdNum, (gcds.get(gcdNum) || 0) + 1);
16        }
17
18        return ans;
19    }
20
21    gcd(a, b) {
22        if (b === 0) {
23            return a;
24        } else {
25            return this.gcd(b, a % b);
26        }
27    }
28 }
```

C++

```
1
2 cpp
3 #include <unordered_map>
4
5 class Solution {
6 public:
7     long long countPairs(vector<int>& nums, int k) {
8         long ans = 0;
9         unordered_map<int, int> gcds;
10
11        for (const int num : nums) {
12            const int gcdNum = gcd(num, k);
13            for (const auto& [gcdVal, count] : gcds) {
14                if (gcdNum * gcdVal % k == 0) {
15                    ans += count;
16                }
17            }
18            ++gcds[gcdNum];
19        }
20
21        return ans;
22    }
23
24 private:
25     int gcd(int a, int b) {
26         if (b == 0) {
27             return a;
28         } else {
29             return gcd(b, a % b);
30         }
31     }
32 };
```

C#

```
1
2 csharp
3 using System.Collections.Generic;
4
5 public class Solution {
6     public long CountPairs(int[] nums, int k) {
7         long ans = 0;
8         Dictionary<int, int> gcds = new Dictionary<int, int>();
9
10        foreach (int num in nums) {
11            int gcdNum = GCD(num, k);
12            foreach (KeyValuePair<int, int> entry in gcds) {
13                if (gcdNum * entry.Key % k == 0) {
14                    ans += entry.Value;
15                }
16            }
17            if (gcds.ContainsKey(gcdNum)) {
18                gcds[gcdNum]++;
19            } else {
20                gcds[gcdNum] = 1;
21            }
22        }
23
24        return ans;
25    }
26
27    private int GCD(int a, int b) {
28        if (b == 0) {
29            return a;
30        } else {
31            return GCD(b, a % b);
32        }
33    }
34 }
```

In conclusion, the above solutions follow the high-level approach explained earlier and find the count of pairs such that their product is divisible by `k`. Each solution utilizes the greatest common divisor (gcd) function to calculate the gcd values, iterate through each pair of gcd values, and check if their product is divisible by `k`. The implementation is provided in Python, Java, JavaScript, C++, and C#.



Level Up Your
Algo Skills

Get Premium