



DEPARTMENT OF COMPUTERSCIENCE & ENGINEERING

Discover. Learn. Empower.

Assignment - 10

Student Name: Gaurav Saini

Branch: BE-CSE

Semester:06

UID: 22BCS11085

Section/Group:NTPP_IOT_603_B

Date of Performance:07-04-2025

Subject Name: AP LAB-II

Subject Code: 22CSP-351

1. **Aim:**
 - a. **Hamming Distance**
 - b. **No. of 1 Bits**
 - c. **Divide Two Integers**
2. **Implementation/Code:**

A. Hamming Distance

```
class Solution {  
  
public:  
  
    int hammingDistance(int x, int y) {  
  
        int xorResult = x ^ y;  
  
        int bitDifference = 0;  
  
        while (xorResult) {  
  
            bitDifference += xorResult & 1;  
  
            xorResult >>= 1;  
  
        }  
    }  
}
```



```
        return bitDifference;
    }
};
```

B. No. of 1 Bits

```
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n != 0) {
            count += (n & 1);
            n >>= 1;
        }
        return count;
    }
};
```

C. Divide Two Integers

```
class Solution {
public:
    int divide(int dividend, int divisor) {

        if (dividend == INT_MIN && divisor == -1) {
            return INT_MAX;
        }

        long long absDividend = abs((long long)dividend);
        long long absDivisor = abs((long long)divisor);

        long long quotient = 0;

        while (absDividend >= absDivisor) {
            long long tempDivisor = absDivisor, multiple = 1;

            while (absDividend >= (tempDivisor << 1)) {
                tempDivisor <<= 1;
                multiple <<= 1;
            }

            absDividend -= tempDivisor;
            quotient += multiple;
        }

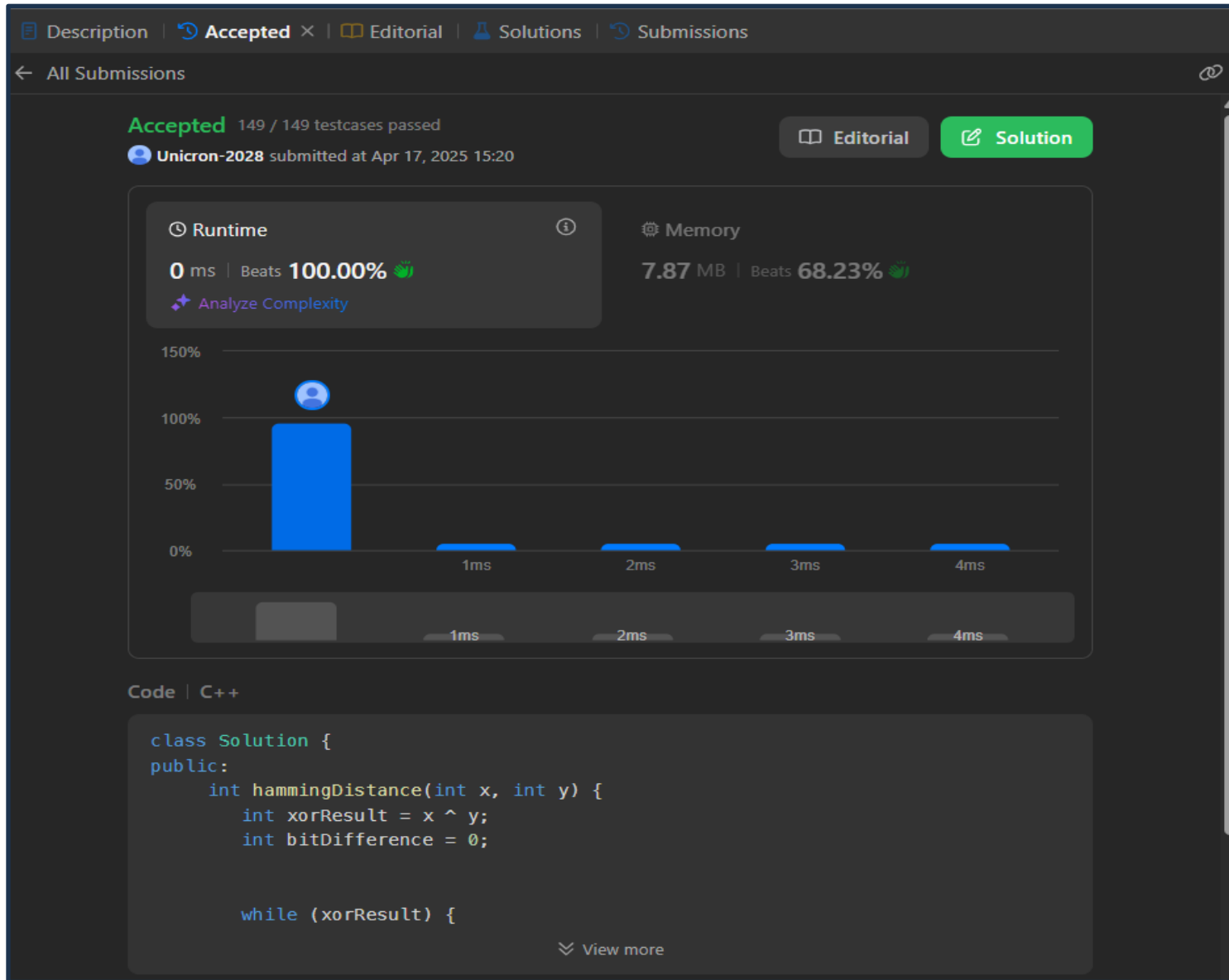
        if ((dividend < 0) ^ (divisor < 0)) {
            quotient = -quotient;
        }
        if (quotient > INT_MAX) return INT_MAX;
        if (quotient < INT_MIN) return INT_MIN;

        return quotient;
    }
};
```



3. Output

A. Hamming Distance





DEPARTMENT OF COMPUTERSCIENCE & ENGINEERING

Discover. Learn. Empower.

B. No. of 1 Bits

[Description](#) | [Accepted](#) × | [Editorial](#) | [Solutions](#) | [Submissions](#)

[← All Submissions](#)

Accepted 598 / 598 testcases passed

[Unicon-2028](#) submitted at Apr 17, 2025 15:20

[Editorial](#) [Solution](#)

Runtime ⓘ
0 ms | Beats 100.00% 🏆
[Analyze Complexity](#)

Memory
8.32 MB | Beats 11.84%

Runtime	Beats
0 ms	100.00%
1 ms	
2 ms	
3 ms	
4 ms	

Code | C++

```
class Solution {
public:
    int hammingWeight(int n) {

        int count = 0;
        while (n != 0) {
            count += (n & 1);
            n >>= 1;
        }
    }
};
```

[View more](#)



DEPARTMENT OF COMPUTERSCIENCE & ENGINEERING

Discover. Learn. Empower.

C. Divide Two Integers

[Description](#) | [Accepted](#) × | [Editorial](#) | [Solutions](#) | [Submissions](#)

← All Submissions

Accepted 994 / 994 testcases passed

Unicon-2028 submitted at Apr 17, 2025 15:21

Editorial

Solution

Runtime

1 ms | Beats 14.11%

Analyze Complexity

Memory

8.60 MB | Beats 71.83%

7ms 157ms 307ms 457ms 606ms 756ms 906ms 1056ms

Code | C++

```
class Solution {
public:
    int divide(int dividend, int divisor) {
        // Handle edge case where the result might overflow
        if (dividend == INT_MIN && divisor == -1) {
            return INT_MAX; // Overflow condition
        }
    }
}
```

View more

More challenges



DEPARTMENT OF COMPUTERSCIENCE & ENGINEERING

Discover. Learn. Empower.