CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

AP ASSIGNMENT 4

ALIZA ASIF
22BCS50175
22BCS_FL_IOT_601_A

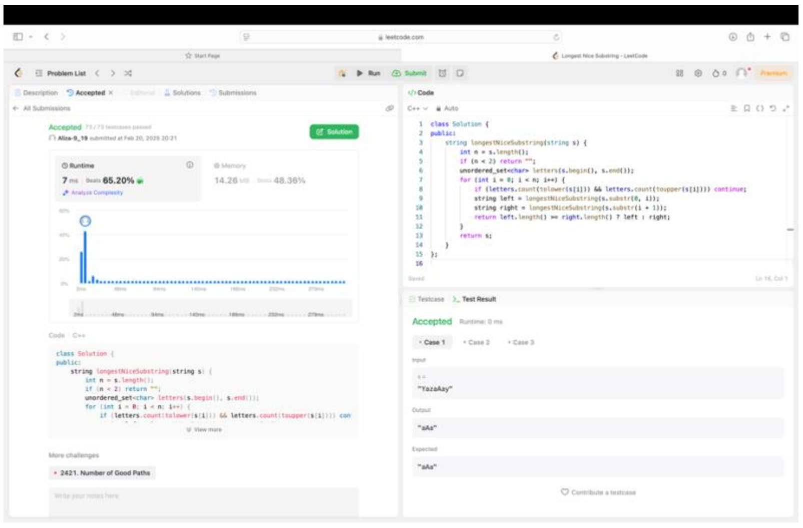
AP ASSIGNMENT 4

Q1. LONGEST NICE SUBSTRING (1763)

Implementation Code:

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n = s.length();
        if (n < 2) return "";
        unordered_set<char> letters(s.begin(), s.end());
        for (int i = 0; i < n; i++) {
            if (letters.count(tolower(s[i])) && letters.count(toupper(s[i]))) continue;
            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));
            return left.length() >= right.length() ? left : right;
        }
        return s;
    }
};
```

Output:



Accepted 73 / 73 testcases passed
Aliza-9_19 submitted at Feb 20, 2026 20:21

Runtime
7 ms | Beats 65.20%

Memory
14.26 MB | Beats 48.36%

```
class Solution {
public:
    string longestNiceSubstring(string s) {
        int n = s.length();
        if (n < 2) return "";
        unordered_set<char> letters(s.begin(), s.end());
        for (int i = 0; i < n; i++) {
            if (letters.count(tolower(s[i])) && letters.count(toupper(s[i]))) continue;
            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));
            return left.length() >= right.length() ? left : right;
        }
        return s;
    }
};
```

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

s =
"YazaAay"

Output

"aAa"

Expected

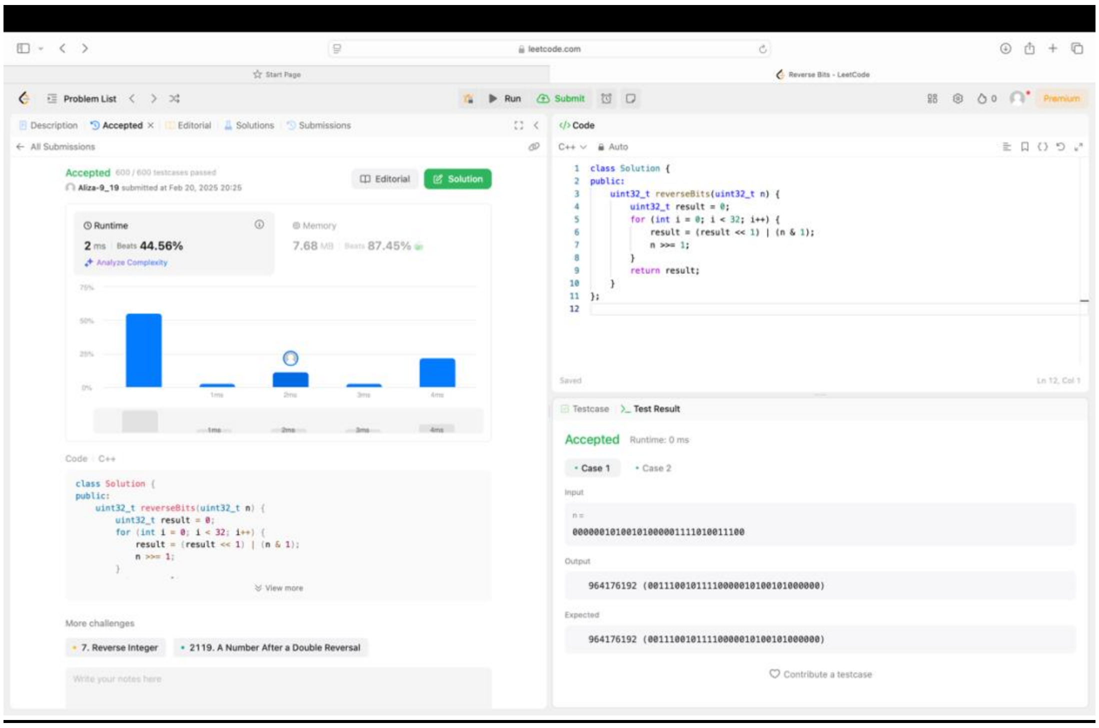
"aAa"

Q2. REVERSE BITS (190)

Implementation Code:

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            result = (result << 1) | (n & 1);
            n >>= 1;
        }
        return result;
    }
};
```

Output:



Accepted 600 / 600 testcases passed
Aliza-9_19 submitted at Feb 20, 2025 20:25

Runtime
2 ms | Beats 44.56%
Analyze Complexity

Memory
7.68 MB | Beats 87.45%

Code | C++

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            result = (result << 1) | (n & 1);
            n >>= 1;
        }
```

View more

More challenges
- 7. Reverse Integer
- 2119. A Number After a Double Reversal

Write your notes here

```
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            result = (result << 1) | (n & 1);
            n >>= 1;
        }
        return result;
    }
};
```

Saved | Ln 12, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

```
n =
00000010100101000001111010011100
```

Output

```
964176192 (00111001011110000010100101000000)
```

Expected

```
964176192 (00111001011110000010100101000000)
```

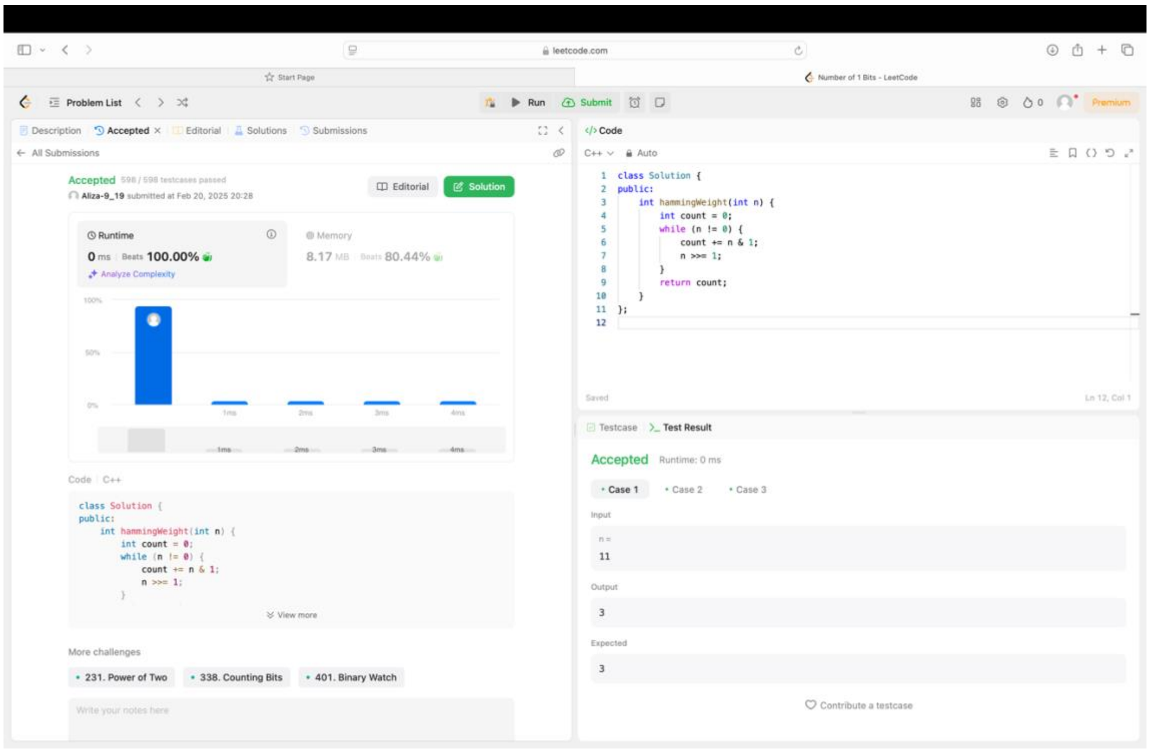
Contribute a testcase

Q3. Number of 1 Bits (191)

Implementation Code:

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

Output:



Accepted 598 / 598 testcases passed

Aliza-9_19 submitted at Feb 20, 2025 20:28

Runtime
0 ms | Beats 100.00%

Memory
8.17 MB | Beats 80.44%

Code | C++

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

More challenges

- 231. Power of Two
- 338. Counting Bits
- 401. Binary Watch

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

Accepted Runtime: 0 ms

Input

n =
11

Output

3

Expected

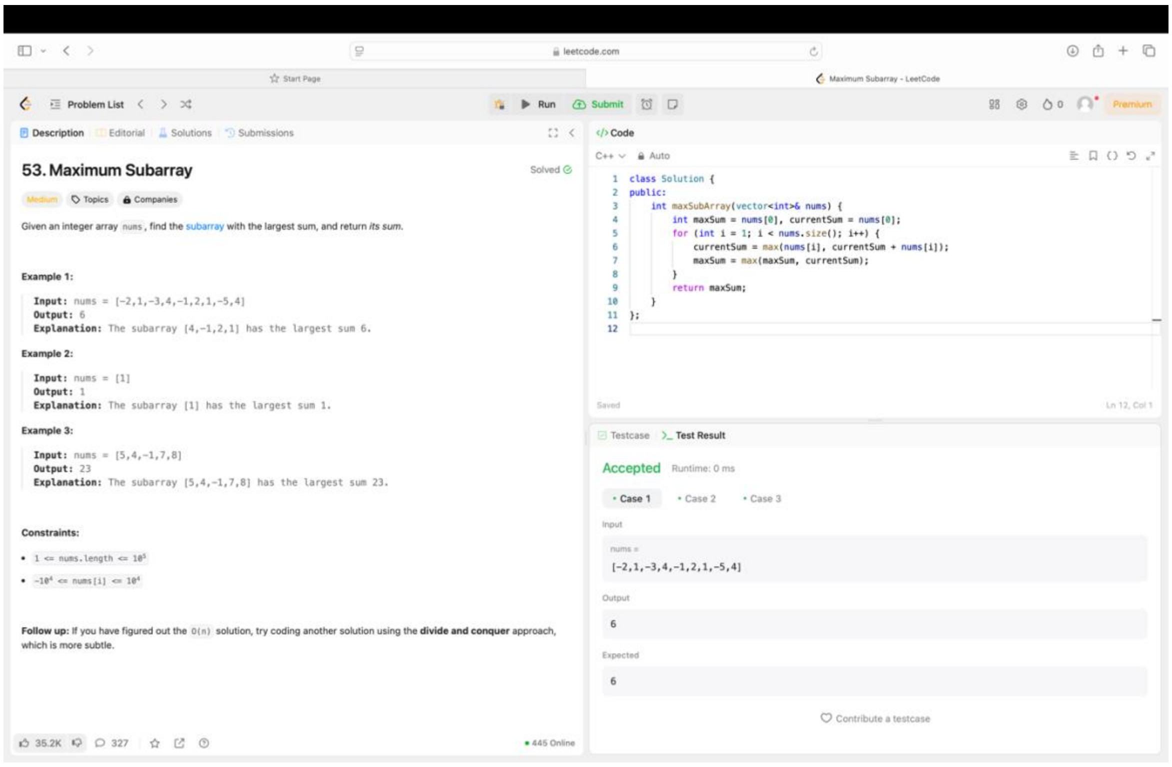
3

Q4. Maximum Subarray (53)

Implementation Code:

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0], currentSum = nums[0];
        for (int i = 1; i < nums.size(); i++) {
            currentSum = max(nums[i], currentSum + nums[i]);
            maxSum = max(maxSum, currentSum);
        }
        return maxSum;
    }
};
```

Output:



53. Maximum Subarray

Solved

Medium Topics Companies

Given an integer array `nums`, find the subarray with the largest sum, and return *its sum*.

Example 1:

```
Input: nums = [-2,1,-3,4,-1,2,1,-5,4]
Output: 6
Explanation: The subarray [4,-1,2,1] has the largest sum 6.
```

Example 2:

```
Input: nums = [1]
Output: 1
Explanation: The subarray [1] has the largest sum 1.
```

Example 3:

```
Input: nums = [5,4,-1,7,8]
Output: 23
Explanation: The subarray [5,4,-1,7,8] has the largest sum 23.
```

Constraints:

- `1 <= nums.length <= 10`^{5}
- `-10`^{4}` <= nums[i] <= 10`^{4}

Follow up: If you have figured out the `O(n)` solution, try coding another solution using the **divide and conquer** approach, which is more subtle.

35.2K 327 • 445 Online

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0], currentSum = nums[0];
        for (int i = 1; i < nums.size(); i++) {
            currentSum = max(nums[i], currentSum + nums[i]);
            maxSum = max(maxSum, currentSum);
        }
        return maxSum;
    }
};
```

Saved Ln 12, Col 1

Testcase Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

```
nums =
[-2,1,-3,4,-1,2,1,-5,4]
```

Output

```
6
```

Expected

```
6
```

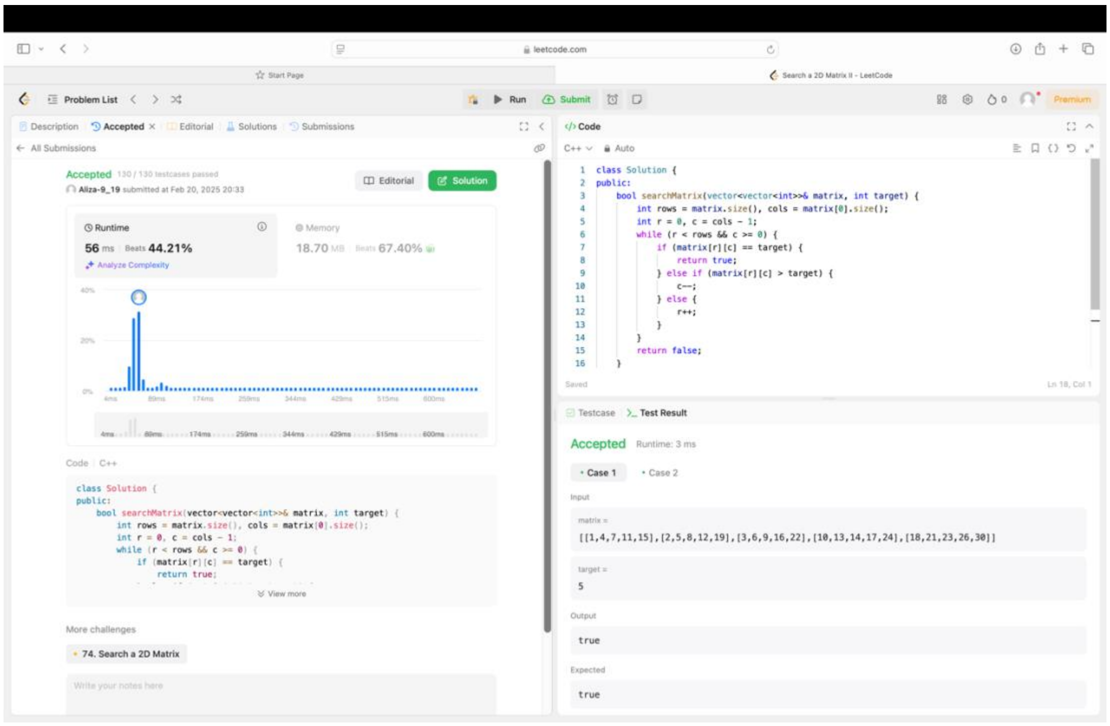
Contribute a testcase

Q5. Search a 2D Matrix II (240)

Implementation Code:

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int rows = matrix.size(), cols = matrix[0].size();
        int r = 0, c = cols - 1;
        while (r < rows && c >= 0) {
            if (matrix[r][c] == target) {
                return true;
            } else if (matrix[r][c] > target) {
                c--;
            } else {
                r++;
            }
        }
        return false;
    }
};
```

Output:



Accepted 130 / 130 testcases passed
Aliza-9_19 submitted at Feb 20, 2025 20:33

Runtime
56 ms | Beats 44.21%

Memory
18.70 MB | Beats 67.40%

Code | C++

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int rows = matrix.size(), cols = matrix[0].size();
        int r = 0, c = cols - 1;
        while (r < rows && c >= 0) {
            if (matrix[r][c] == target) {
                return true;
```

More challenges
74. Search a 2D Matrix

Write your notes here

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        int rows = matrix.size(), cols = matrix[0].size();
        int r = 0, c = cols - 1;
        while (r < rows && c >= 0) {
            if (matrix[r][c] == target) {
                return true;
            } else if (matrix[r][c] > target) {
                c--;
            } else {
                r++;
            }
        }
        return false;
    }
```

Accepted Runtime: 3 ms

Case 1 Case 2

Input

```
matrix =
[[1,4,7,11,15],[2,5,8,12,19],[3,6,9,16,22],[10,13,14,17,24],[18,21,23,26,30]]

target =
5
```

Output

```
true
```

Expected

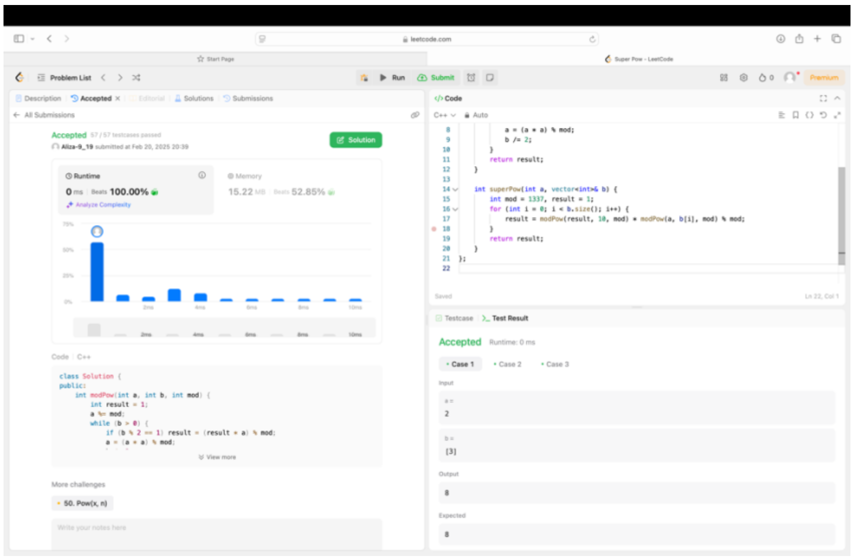
```
true
```

Q6. Super Pow (372)

Implementation Code:

```
class Solution {
public:
    int modPow(int a, int b, int mod) {
        int result = 1;
        a %= mod;
        while (b > 0) {
            if (b % 2 == 1) result = (result * a) % mod;
            a = (a * a) % mod;
            b /= 2;
        }
        return result;
    }
    int superPow(int a, vector<int>& b) {
        int mod = 1337, result = 1;
        for (int i = 0; i < b.size(); i++) {
            result = modPow(result, 10, mod) * modPow(a, b[i], mod) % mod;
        }
        return result;
    }
};
```

Output:



leetcode.com

Start Page | Super Pow - LeetCode

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

← All Submissions

Accepted 57 / 57 testcases passed

Aliza-9_19 submitted at Feb 20, 2025 20:39

Solution

Runtime 0 ms | Beats 100.00%

Analyze Complexity

Memory 15.22 MB | Beats 52.85%

Code | C++

```
class Solution {
public:
    int modPow(int a, int b, int mod) {
        int result = 1;
        a %= mod;
        while (b > 0) {
            if (b % 2 == 1) result = (result * a) % mod;
            a = (a * a) % mod;
```

View more

More challenges

50. Pow(x, n)

Write your notes here

Code

C++ | Auto

```
            a = (a * a) % mod;
            b /= 2;
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        int mod = 1337, result = 1;
        for (int i = 0; i < b.size(); i++) {
            result = modPow(result, 10, mod) * modPow(a, b[i], mod) % mod;
        }
        return result;
    }
};
```

Saved | Ln 22, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2 | Case 3

Input

a =
2

b =
[3]

Output

8

Expected

8