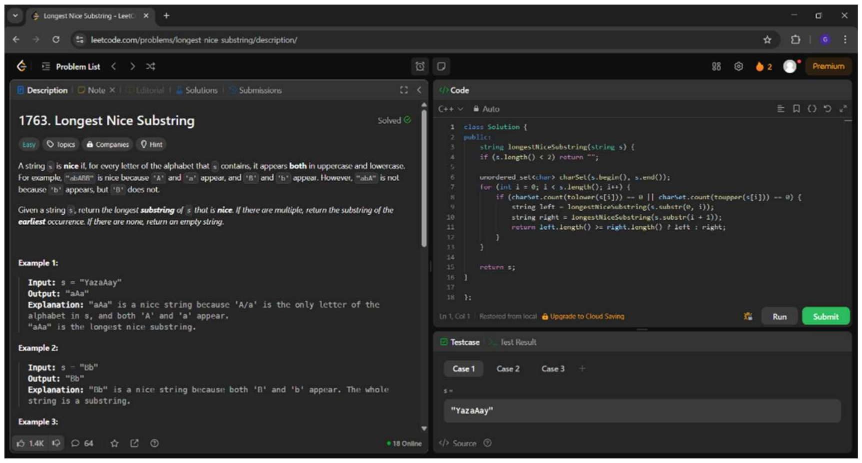
Longest Nice Substring:



leetcode.com/problems/longest-nice-substring/description/

1763. Longest Nice Substring

Solved

Easy · Topics · Companies · Hint

A string `s` is **nice** if, for every letter of the alphabet that `s` contains, it appears **both** in uppercase and lowercase. For example, `"abABB"` is nice because `'A'` and `'a'` appear, and `'B'` and `'b'` appear. However, `"abA"` is not because `'b'` appears, but `'B'` does not.

Given a string `s`, return *the longest* ***substring*** *of* `s` *that is* ***nice****. If there are multiple, return the substring of the* ***earliest*** *occurrence. If there are none, return an empty string.*

Example 1:

```
Input: s = "YazaAay"
Output: "aAa"
Explanation: "aAa" is a nice string because 'A/a' is the only letter of the
alphabet in s, and both 'A' and 'a' appear.
"aAa" is the longest nice substring.
```

Example 2:

```
Input: s = "Bb"
Output: "Bb"
Explanation: "Bb" is a nice string because both 'B' and 'b' appear. The whole
string is a substring.
```

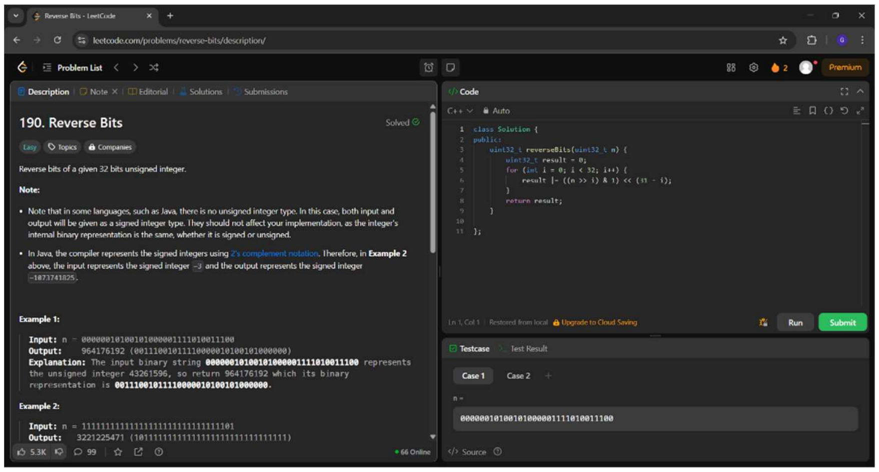
Example 3:

```cpp
class Solution {
public:
    string longestNiceSubstring(string s) {
    if (s.length() < 2) return "";

    unordered_set<char> charSet(s.begin(), s.end());
    for (int i = 0; i < s.length(); i++) {
        if (charSet.count(tolower(s[i])) == 0 || charSet.count(toupper(s[i])) == 0) {
            string left = longestNiceSubstring(s.substr(0, i));
            string right = longestNiceSubstring(s.substr(i + 1));
            return left.length() >= right.length() ? left : right;
        }
    }

    return s;
}

};
```

Testcase: Case 1 · Case 2 · Case 3

s =
"YazaAay"

Reverse Bits:



leetcode.com/problems/reverse-bits/description/

190. Reverse Bits

Solved

Easy · Topics · Companies

Reverse bits of a given 32 bits unsigned integer.

Note:

- Note that in some languages, such as Java, there is no unsigned integer type. In this case, both input and output will be given as a signed integer type. They should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
- In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in **Example 2** above, the input represents the signed integer `-3` and the output represents the signed integer `-1073741825`.

Example 1:

```
Input: n = 00000010100101000001111010011100
Output:    964176192 (00111001011110000010100101000000)
Explanation: The input binary string 00000010100101000001111010011100 represents
the unsigned integer 43261596, so return 964176192 which its binary
representation is 00111001011110000010100101000000.
```

Example 2:

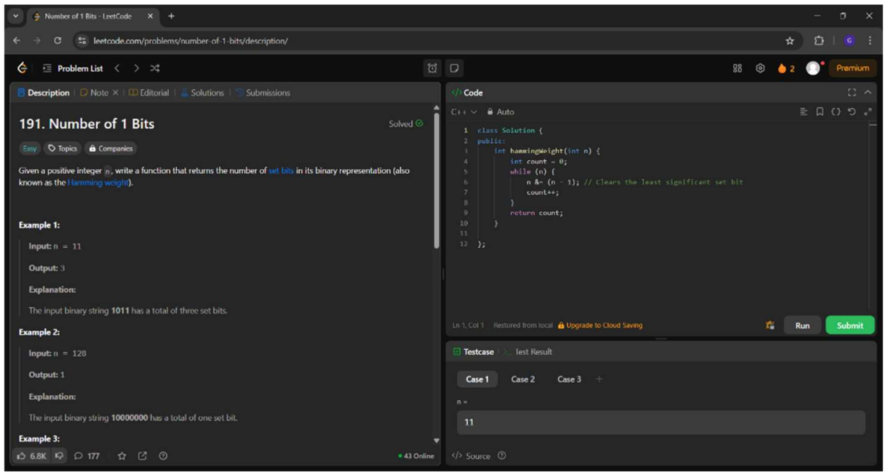
```
Input: n = 11111111111111111111111111111101
Output:   3221225471 (10111111111111111111111111111111)
```

```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        uint32_t result = 0;
        for (int i = 0; i < 32; i++) {
            result |= ((n >> i) & 1) << (31 - i);
        }
        return result;
    }

};
```

Testcase: Case 1 · Case 2

n =
00000010100101000001111010011100

Number of 1 Bits:



leetcode.com/problems/number-of-1-bits/description/

191. Number of 1 Bits

Solved

Easy · Topics · Companies

Given a positive integer `n`, write a function that returns the number of set bits in its binary representation (also known as the Hamming weight).

Example 1:

Input: n = 11

Output: 3

Explanation:

The input binary string **1011** has a total of three set bits.

Example 2:

Input: n = 128

Output: 1

Explanation:

The input binary string **10000000** has a total of one set bit.

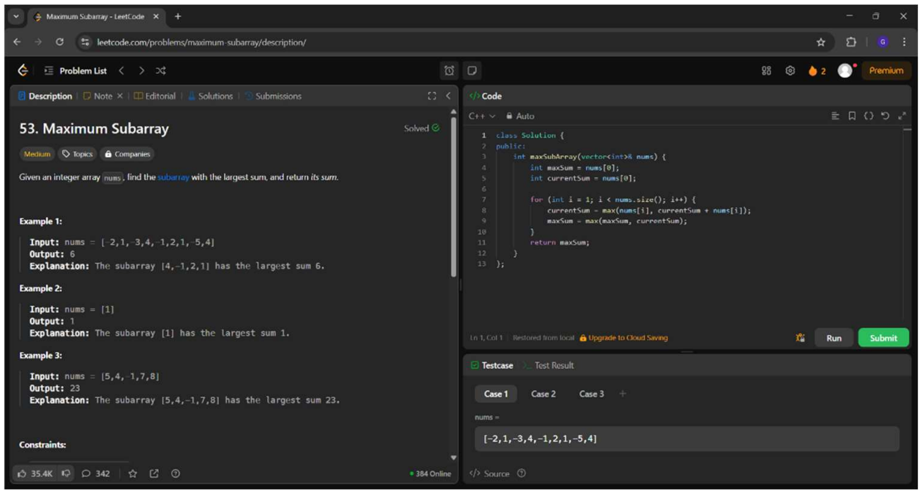
Example 3:

```cpp
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n) {
            n &= (n - 1); // Clears the least significant set bit
            count++;
        }
        return count;
    }

};
```

Testcase: Case 1 · Case 2 · Case 3

n =
11

Maximum Subarray:



leetcode.com/problems/maximum-subarray/description/

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:

35.4K · 342 · 384 Online

Code · C++ · Auto

```cpp
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int maxSum = nums[0];
        int currentSum = nums[0];

        for (int i = 1; i < nums.size(); i++) {
            currentSum = max(nums[i], currentSum + nums[i]);
            maxSum = max(maxSum, currentSum);
        }
        return maxSum;
    }
};
```

Ln 1, Col 1 · Restored from local · Upgrade to Cloud Saving · Run · Submit

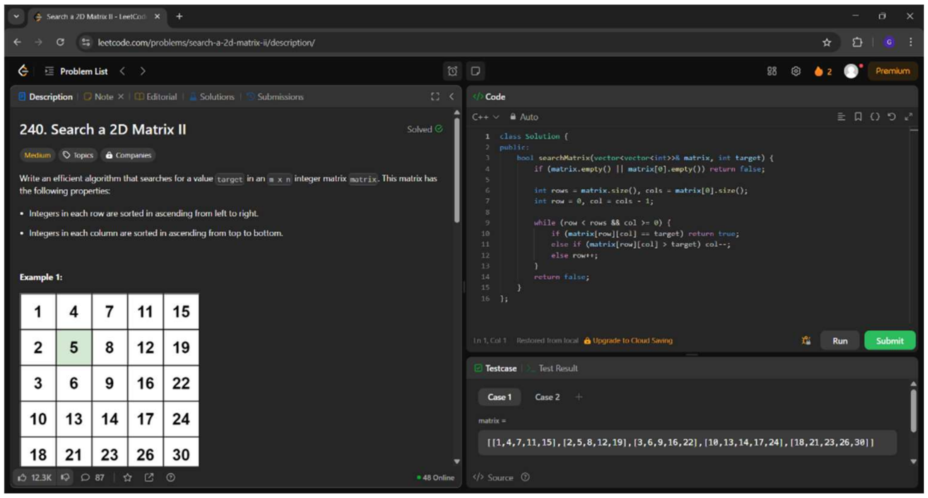
Testcase · Test Result

Case 1 · Case 2 · Case 3

nums =

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

Search a 2D Matrix II:



leetcode.com/problems/search-a-2d-matrix-ii/description/

240. Search a 2D Matrix II

Solved

Medium · Topics · Companies

Write an efficient algorithm that searches for a value `target` in an `m x n` integer matrix `matrix`. This matrix has the following properties:

- Integers in each row are sorted in ascending from left to right.
- Integers in each column are sorted in ascending from top to bottom.

Example 1:

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

12.3K · 87 · 48 Online

Code · C++ · Auto

```cpp
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix, int target) {
        if (matrix.empty() || matrix[0].empty()) return false;

        int rows = matrix.size(), cols = matrix[0].size();
        int row = 0, col = cols - 1;

        while (row < rows && col >= 0) {
            if (matrix[row][col] == target) return true;
            else if (matrix[row][col] > target) col--;
            else row++;
        }
        return false;
    }
};
```

Ln 1, Col 1 · Restored from local · Upgrade to Cloud Saving · Run · Submit

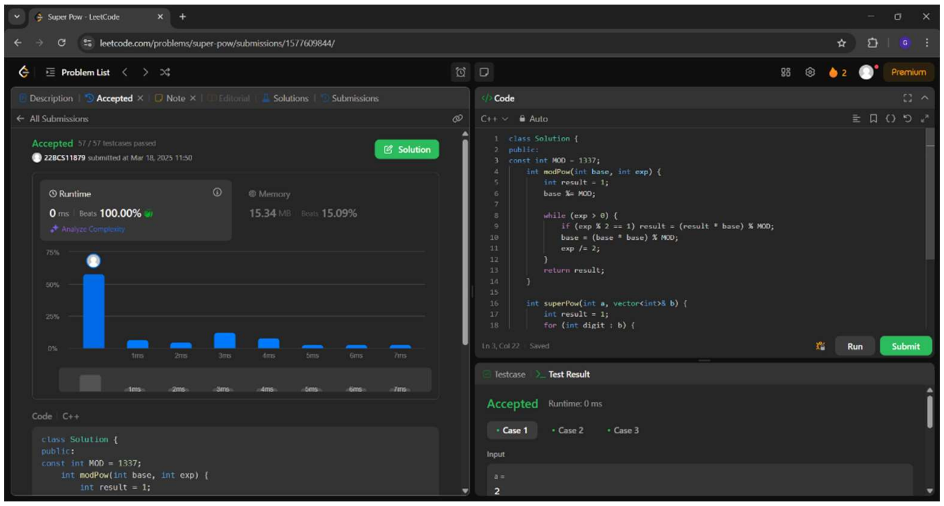
Testcase · Test Result

Case 1 · Case 2

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]]
```

Super Pow:



leetcode.com/problems/super-pow/submissions/1577609844/

Description · Accepted · Note · Editorial · Solutions · Submissions

All Submissions

Accepted 57 / 57 testcases passed

22BCS11879 submitted at Mar 18, 2025 11:50

Solution

Runtime: 0 ms | Beats 100.00%

Analyze Complexity

Memory: 15.34 MB | Beats 15.09%

Code | C++

```cpp
class Solution {
public:
const int MOD = 1337;
    int modPow(int base, int exp) {
        int result = 1;
        base %= MOD;

        while (exp > 0) {
            if (exp % 2 == 1) result = (result * base) % MOD;
            base = (base * base) % MOD;
            exp /= 2;
        }
        return result;
    }

    int superPow(int a, vector<int>& b) {
        int result = 1;
        for (int digit : b) {
```

Ln 3, Col 22 · Saved · Run · Submit

Testcase · Test Result

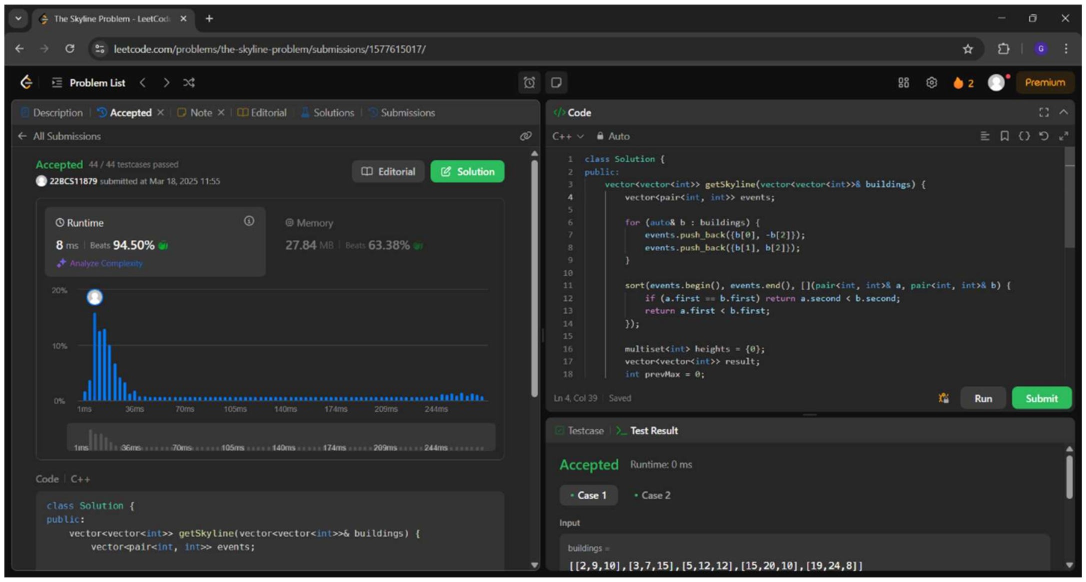
Accepted Runtime: 0 ms

Case 1 · Case 2 · Case 3

Input

```
a =
2
```

Skyline Problem:



leetcode.com/problems/the-skyline-problem/submissions/1577615017/

Accepted 44 / 44 testcases passed

22BCS11879 submitted at Mar 18, 2025 11:55

Runtime 8 ms | Beats 94.50%

Memory 27.84 MB | Beats 63.38%

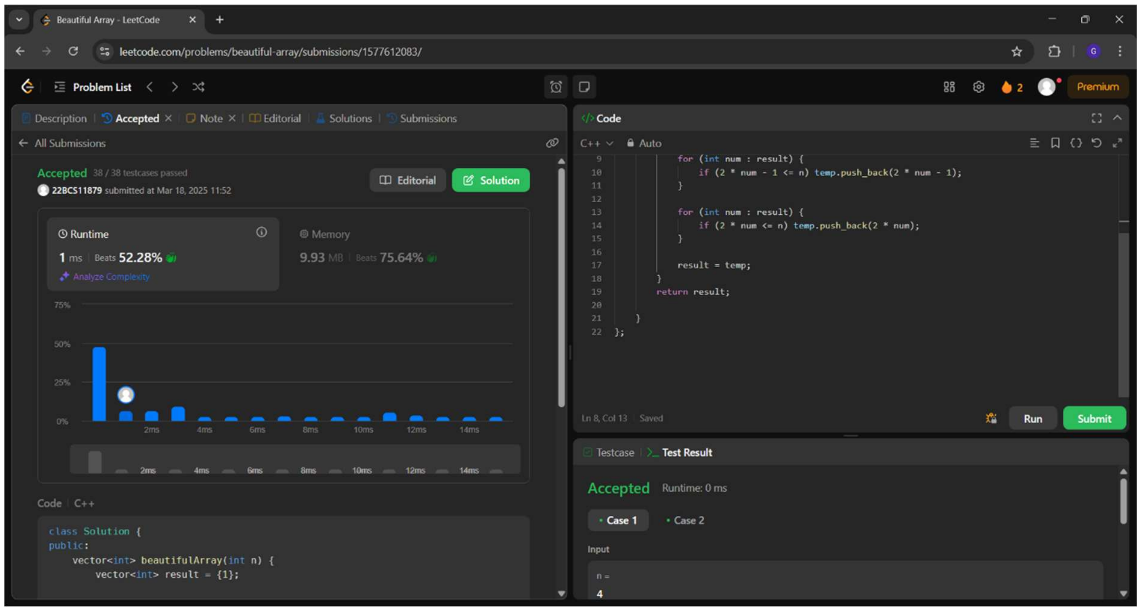
```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
        vector<pair<int, int>> events;

        for (auto& b : buildings) {
            events.push_back({b[0], -b[2]});
            events.push_back({b[1], b[2]});
        }

        sort(events.begin(), events.end(), [](pair<int, int>& a, pair<int, int>& b) {
            if (a.first == b.first) return a.second < b.second;
            return a.first < b.first;
        });

        multiset<int> heights = {0};
        vector<vector<int>> result;
        int prevMax = 0;
```

Accepted Runtime: 0 ms

Input

```
buildings =
[[2,9,10],[3,7,15],[5,12,12],[15,20,10],[19,24,8]]
```

Beautiful Array:



leetcode.com/problems/beautiful-array/submissions/1577612083/

Accepted 38 / 38 testcases passed

22BCS11879 submitted at Mar 18, 2025 11:52

Runtime 1 ms | Beats 52.28%

Memory 9.93 MB | Beats 75.64%

```
class Solution {
public:
    vector<int> beautifulArray(int n) {
        vector<int> result = {1};
```

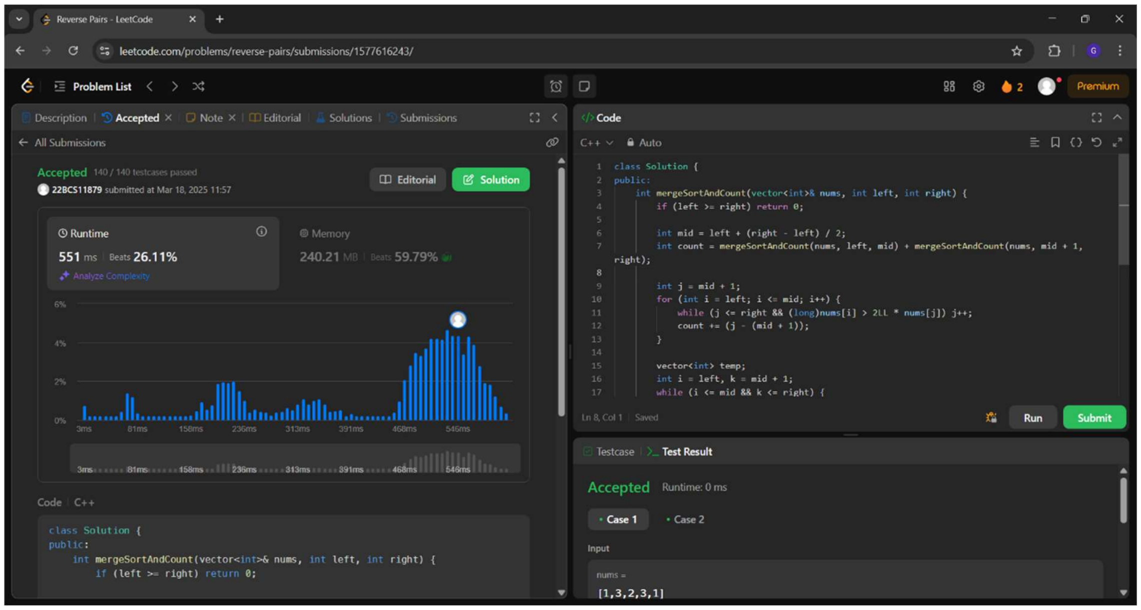
```
            for (int num : result) {
                if (2 * num - 1 <= n) temp.push_back(2 * num - 1);
            }

            for (int num : result) {
                if (2 * num <= n) temp.push_back(2 * num);
            }

            result = temp;
        }
        return result;

    }
};
```

Accepted Runtime: 0 ms

Input

```
n =
4
```

Reverse Pairs:


Reverse Pairs - LeetCode

leetcode.com/problems/reverse-pairs/submissions/1577616243/

Accepted 140 / 140 testcases passed

22BCS11879 submitted at Mar 18, 2025 11:57

Runtime 551 ms | Beats 26.11%

Memory 240.21 MB | Beats 59.79%

```
class Solution {
public:
    int mergeSortAndCount(vector<int>& nums, int left, int right) {
        if (left >= right) return 0;

        int mid = left + (right - left) / 2;
        int count = mergeSortAndCount(nums, left, mid) + mergeSortAndCount(nums, mid + 1,
right);

        int j = mid + 1;
        for (int i = left; i <= mid; i++) {
            while (j <= right && (long)nums[i] > 2LL * nums[j]) j++;
            count += (j - (mid + 1));
        }

        vector<int> temp;
        int i = left, k = mid + 1;
        while (i <= mid && k <= right) {
```

Ln 8, Col 1 | Saved

Accepted Runtime: 0 ms

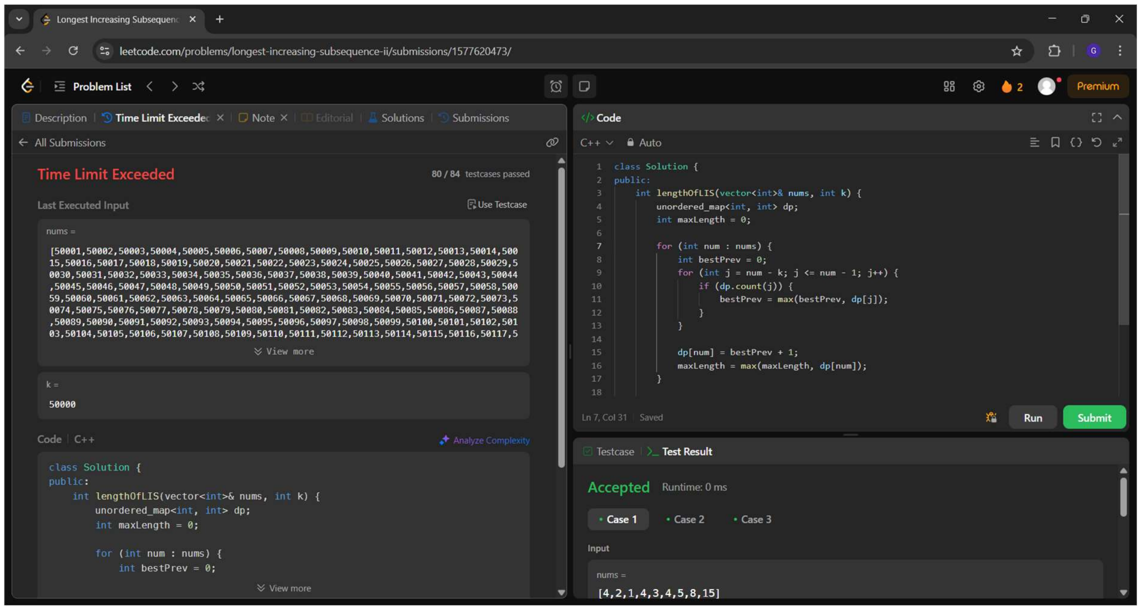
Case 1 Case 2

Input

nums =

```
[1,3,2,3,1]
```

Longest Increasing Subsequence II:


Longest Increasing Subsequence

leetcode.com/problems/longest-increasing-subsequence-ii/submissions/1577620473/

Time Limit Exceeded 80 / 84 testcases passed

Last Executed Input

nums =

```
[50001,50002,50003,50004,50005,50006,50007,50008,50009,50010,50011,50012,50013,50014,500
15,50016,50017,50018,50019,50020,50021,50022,50023,50024,50025,50026,50027,50028,50029,5
0030,50031,50032,50033,50034,50035,50036,50037,50038,50039,50040,50041,50042,50043,50044
,50045,50046,50047,50048,50049,50050,50051,50052,50053,50054,50055,50056,50057,50058,500
59,50060,50061,50062,50063,50064,50065,50066,50067,50068,50069,50070,50071,50072,50073,5
0074,50075,50076,50077,50078,50079,50080,50081,50082,50083,50084,50085,50086,50087,50088
,50089,50090,50091,50092,50093,50094,50095,50096,50097,50098,50099,50100,50101,50102,501
03,50104,50105,50106,50107,50108,50109,50110,50111,50112,50113,50114,50115,50116,50117,5
```

k =

```
50000
```

```
class Solution {
public:
    int lengthOfLIS(vector<int>& nums, int k) {
        unordered_map<int, int> dp;
        int maxLength = 0;

        for (int num : nums) {
            int bestPrev = 0;
            for (int j = num - k; j <= num - 1; j++) {
                if (dp.count(j)) {
                    bestPrev = max(bestPrev, dp[j]);
                }
            }

            dp[num] = bestPrev + 1;
            maxLength = max(maxLength, dp[num]);
        }

```

Ln 7, Col 31 | Saved

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

nums =

```
[4,2,1,4,3,4,5,8,15]
```