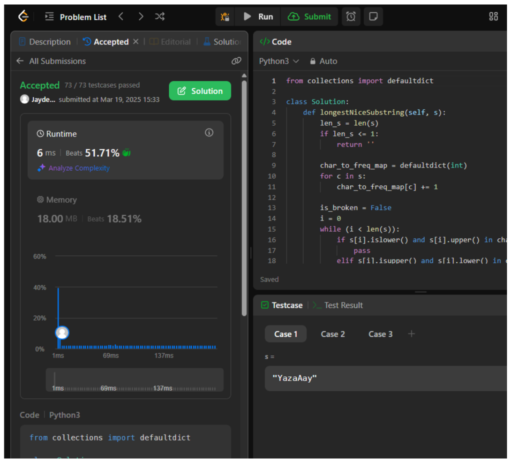
question 1 :
longest nice string :



Accepted 73 / 73 testcases passed
Jayde... submitted at Mar 19, 2025 15:33

Runtime
6 ms | Beats 51.71%
Analyze Complexity

Memory
18.00 MB | Beats 18.51%

```
from collections import defaultdict

class Solution:
    def longestNiceSubstring(self, s):
        len_s = len(s)
        if len_s <= 1:
            return ''

        char_to_freq_map = defaultdict(int)
        for c in s:
            char_to_freq_map[c] += 1

        is_broken = False
        i = 0
        while (i < len(s)):
            if s[i].islower() and s[i].upper() in cha
                pass
            elif s[i].isupper() and s[i].lower() in c
```

s =
"YazaAay"

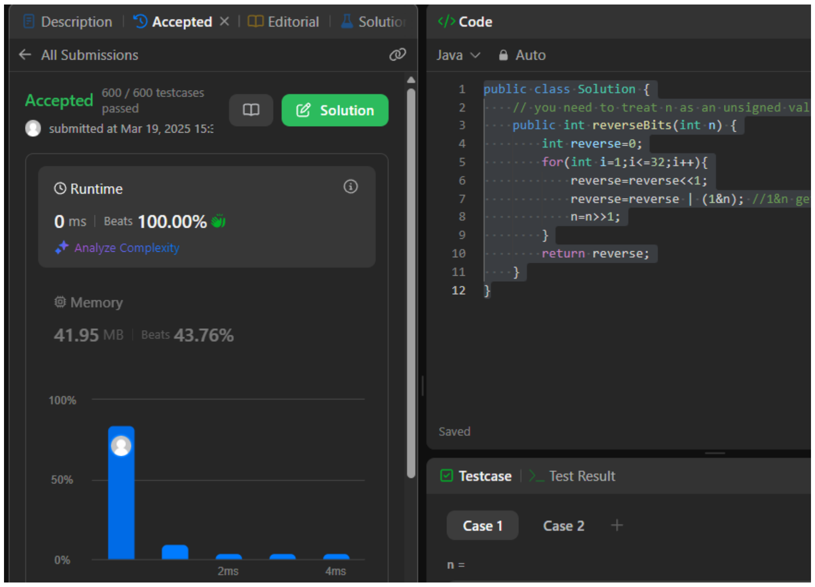
Code | Python3

```
from collections import defaultdict
```

code :

```
from collections import defaultdict



class Solution:
  def longestNiceSubstring(self, s):
    len_s = len(s)
    if len_s <= 1:
      return ''



    char_to_freq_map = defaultdict(int)
    for c in s:
      char_to_freq_map[c] += 1

    is_broken = False
    i = 0
    while (i < len(s)):
      if s[i].islower() and s[i].upper() in char_to_freq_map.keys():
        pass
      elif s[i].isupper() and s[i].lower() in char_to_freq_map.keys():
        pass
      else:
        is_broken = True
        break
      i += 1


    if not is_broken:
      return s

    longest_nice_substr_1 = self.longestNiceSubstring(s[:i])
    longest_nice_substr_2 = self.longestNiceSubstring(s[i+1:])

    if len(longest_nice_substr_1)>=len(longest_nice_substr_2):
      return longest_nice_substr_1
    else:
      return longest_nice_substr_2
```

question 2:
reverse bits :


Description | Accepted × | Editorial | Solutio

All Submissions

Accepted 600 / 600 testcases passed

Solution

submitted at Mar 19, 2025 15:

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

41.95 MB | Beats 43.76%

Code

Java Auto

```
public class Solution {
    // you need to treat n as an unsigned val
    public int reverseBits(int n) {
        int reverse=0;
        for(int i=1;i<=32;i++){
            reverse=reverse<<1;
            reverse=reverse | (1&n); //1&n ge
            n=n>>1;
        }
        return reverse;
    }
}
```

Saved

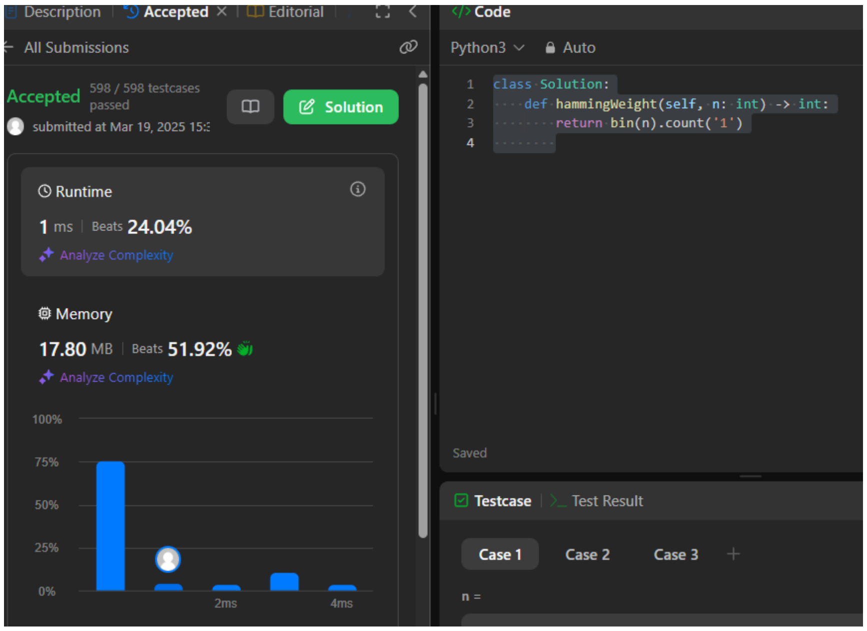
Testcase | Test Result

Case 1 Case 2 +

n =

code :
```
public class Solution {
  // you need to treat n as an unsigned value
  public int reverseBits(int n) {
    int reverse=0;
    for(int i=1;i<=32;i++){
      reverse=reverse<<1;
      reverse=reverse | (1&n); //1&n gets you the rightmost bit
      n=n>>1;
    }
    return reverse;
  }
}
```

question 2:
reverse bits :


Description | Accepted | Editorial

All Submissions

Accepted 598 / 598 testcases passed

submitted at Mar 19, 2025 15:

Solution

Runtime

1 ms | Beats 24.04%

Analyze Complexity

Memory

17.80 MB | Beats 51.92%

Analyze Complexity

Code

Python3 Auto

```
class Solution:
    def hammingWeight(self, n: int) -> int:
        return bin(n).count('1')
        
```

Saved

Testcase | Test Result

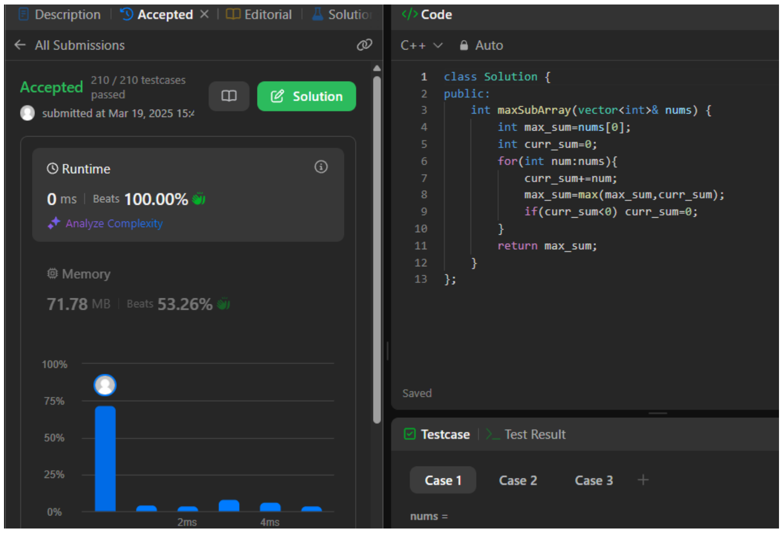
Case 1 Case 2 Case 3

n =

code : class Solution:

```
  def hammingWeight(self, n: int) -> int:
    return bin(n).count('1')



}
```

question 2:
reverse bits :



Accepted 210 / 210 testcases passed

submitted at Mar 19, 2025 15:

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

71.78 MB | Beats 53.26%

```
class Solution {
public:
    int maxSubArray(vector<int>& nums) {
        int max_sum=nums[0];
        int curr_sum=0;
        for(int num:nums){
            curr_sum+=num;
            max_sum=max(max_sum,curr_sum);
            if(curr_sum<0) curr_sum=0;
        }
        return max_sum;
    }
};
```

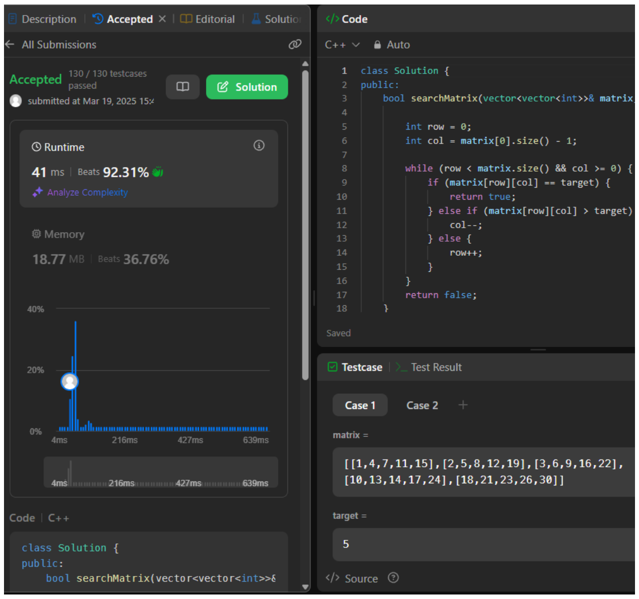
Saved

nums =

code:

```
class Solution {
public:
  int maxSubArray(vector<int>& nums) {
    int max_sum=nums[0];
    int curr_sum=0;
    for(int num:nums){
      curr_sum+=num;
      max_sum=max(max_sum,curr_sum);
      if(curr_sum<0) curr_sum=0;
    }
    return max_sum;
  }
};




}
```

question 2:
reverse bits :



Accepted 130 / 130 testcases passed
submitted at Mar 19, 2025 15:

Runtime
41 ms | Beats 92.31%

Memory
18.77 MB | Beats 36.76%

40%
20%
0%
4ms 216ms 427ms 639ms

Code | C++

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>&
```

```
class Solution {
public:
    bool searchMatrix(vector<vector<int>>& matrix

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

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

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]]
```

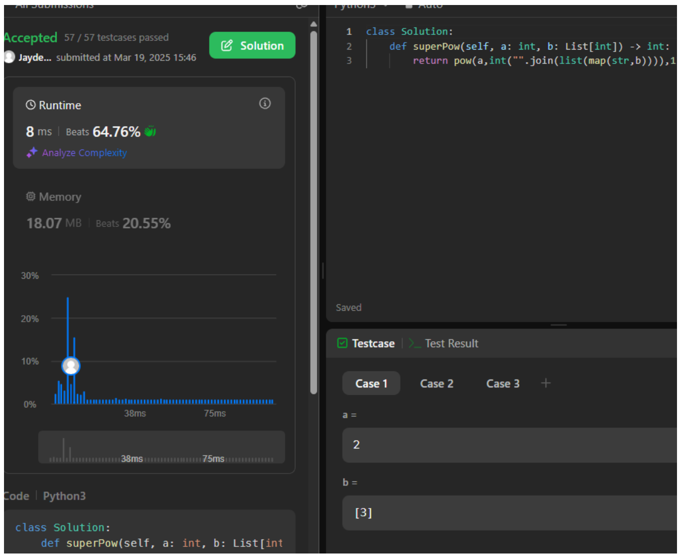
target =

```
5
```

code:

```
class Solution {
public:
  bool searchMatrix(vector<vector<int>>& matrix, int target) {


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


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




}
```

question 2:
reverse bits :



Accepted 57 / 57 testcases passed

Jayde... submitted at Mar 19, 2025 15:46

Runtime

8 ms | Beats 64.76%

Memory

18.07 MB | Beats 20.55%

Code | Python3

```
class Solution:
    def superPow(self, a: int, b: List[int
```

```
class Solution:
    def superPow(self, a: int, b: List[int]) -> int:
        return pow(a,int("".join(list(map(str,b)))),1
```

Saved

Testcase | Test Result

Case 1 Case 2 Case 3

a =

2

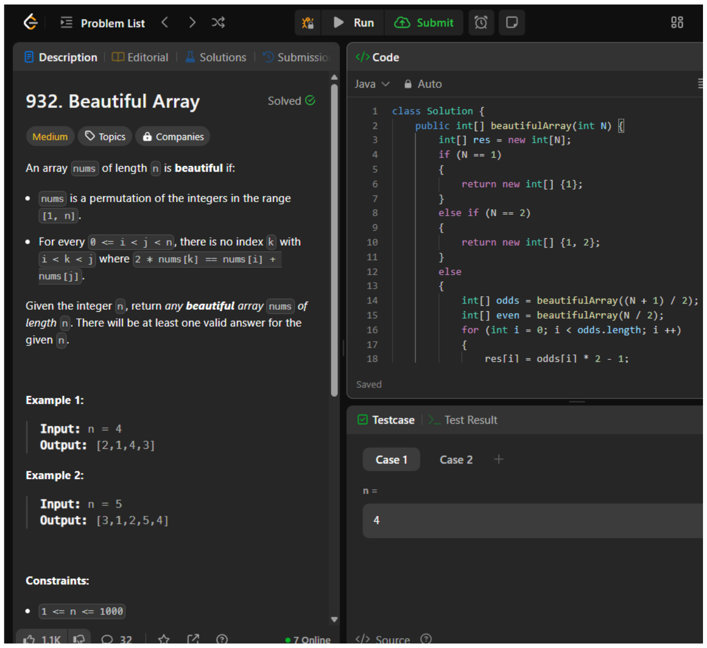
b =

[3]

code:

```
class Solution:
  def superPow(self, a: int, b: List[int]) -> int:
    return pow(a,int("".join(list(map(str,b)))),1337)




}
```

question 2:
reverse bits :



932. Beautiful Array — Solved

An array `nums` of length `n` is **beautiful** if:

- `nums` is a permutation of the integers in the range `[1, n]`.
- For every `0 <= i < j < n`, there is no index `k` with `i < k < j` where `2 * nums[k] == nums[i] + nums[j]`.

Given the integer `n`, return *any* ***beautiful*** *array* `nums` *of length* `n`. There will be at least one valid answer for the given `n`.

Example 1:

```
Input: n = 4
Output: [2,1,4,3]
```

Example 2:

```
Input: n = 5
Output: [3,1,2,5,4]
```

Constraints:

- `1 <= n <= 1000`

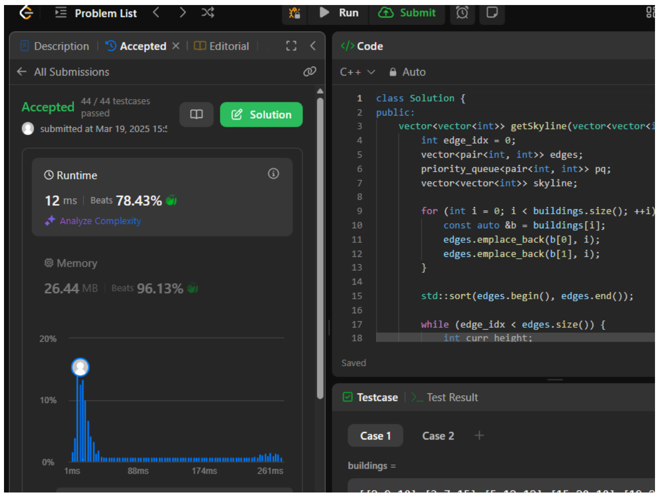
```java
class Solution {
    public int[] beautifulArray(int N) {
        int[] res = new int[N];
        if (N == 1)
        {
            return new int[] {1};
        }
        else if (N == 2)
        {
            return new int[] {1, 2};
        }
        else
        {
            int[] odds = beautifulArray((N + 1) / 2);
            int[] even = beautifulArray(N / 2);
            for (int i = 0; i < odds.length; i ++)
            {
                res[i] = odds[i] * 2 - 1;
```

n =
4

code:

```java
class Solution {
  public int[] beautifulArray(int N) {
    int[] res = new int[N];
    if (N == 1)
    {
      return new int[] {1};
    }
    else if (N == 2)
    {
      return new int[] {1, 2};
    }
    else
    {
      int[] odds = beautifulArray((N + 1) / 2);
      int[] even = beautifulArray(N / 2);
      for (int i = 0; i < odds.length; i ++)
      {
        res[i] = odds[i] * 2 - 1;
      }
      for (int j = 0; j < even.length; j ++)
      {
        res[odds.length + j] = even[j] * 2;
      }
    }
    return res;
  }
}



}
```

question 2:
reverse bits :


Problem List   Run   Submit

Description | Accepted × | Editorial

All Submissions

Accepted 44 / 44 testcases passed

submitted at Mar 19, 2025 15:

Solution

Runtime

12 ms | Beats 78.43%

Analyze Complexity

Memory

26.44 MB | Beats 96.13%

Code

C++   Auto

```
class Solution {
public:
    vector<vector<int>> getSkyline(vector<vector<i
        int edge_idx = 0;
        vector<pair<int, int>> edges;
        priority_queue<pair<int, int>> pq;
        vector<vector<int>> skyline;

        for (int i = 0; i < buildings.size(); ++i)
            const auto &b = buildings[i];
            edges.emplace_back(b[0], i);
            edges.emplace_back(b[1], i);
        }

        std::sort(edges.begin(), edges.end());

        while (edge_idx < edges.size()) {
            int curr_height;
```

Saved

Testcase | Test Result

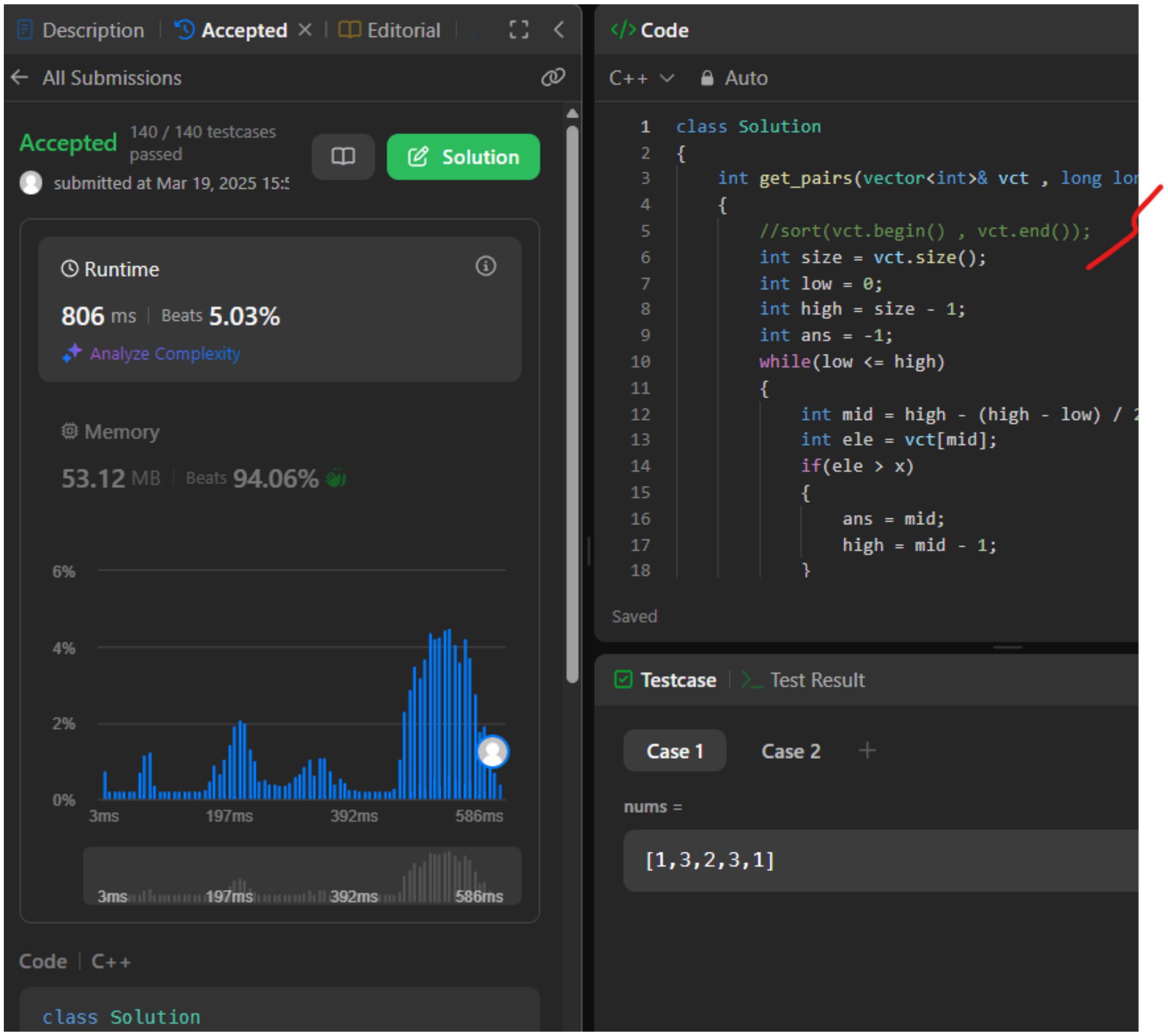
Case 1   Case 2   +

buildings =

code:

```
class Solution {
public:
  vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {
    int edge_idx = 0;
    vector<pair<int, int>> edges;
    priority_queue<pair<int, int>> pq;
    vector<vector<int>> skyline;


    for (int i = 0; i < buildings.size(); ++i) {
      const auto &b = buildings[i];
      edges.emplace_back(b[0], i);
      edges.emplace_back(b[1], i);
    }


    std::sort(edges.begin(), edges.end());


    while (edge_idx < edges.size()) {
      int curr_height;
      const auto &[curr_x, _] = edges[edge_idx];
      while (edge_idx < edges.size() &&
          curr_x == edges[edge_idx].first) {
        const auto &[_, building_idx] = edges[edge_idx];
        const auto &b = buildings[building_idx];
        if (b[0] == curr_x)
          pq.emplace(b[2], b[1]);
        ++edge_idx;
      }
      while (!pq.empty() && pq.top().second <= curr_x)
        pq.pop();
      curr_height = pq.empty() ? 0 : pq.top().first;
      if (skyline.empty() || skyline.back()[1] != curr_height)
        skyline.push_back({curr_x, curr_height});
    }
    return skyline;
  }
};
```

question 2:
reverse bits :


Description | Accepted | Editorial

All Submissions

Accepted 140 / 140 testcases passed

submitted at Mar 19, 2025 15:5

Runtime

806 ms | Beats 5.03%

Analyze Complexity

Memory

53.12 MB | Beats 94.06%

Code | C++

```
class Solution
```

Code

C++ Auto

```
class Solution
{
    int get_pairs(vector<int>& vct , long lo
    {
        //sort(vct.begin() , vct.end());
        int size = vct.size();
        int low = 0;
        int high = size - 1;
        int ans = -1;
        while(low <= high)
        {
            int mid = high - (high - low) / 2
            int ele = vct[mid];
            if(ele > x)
            {
                ans = mid;
                high = mid - 1;
            }
```

Saved

Testcase | Test Result

Case 1 Case 2 +

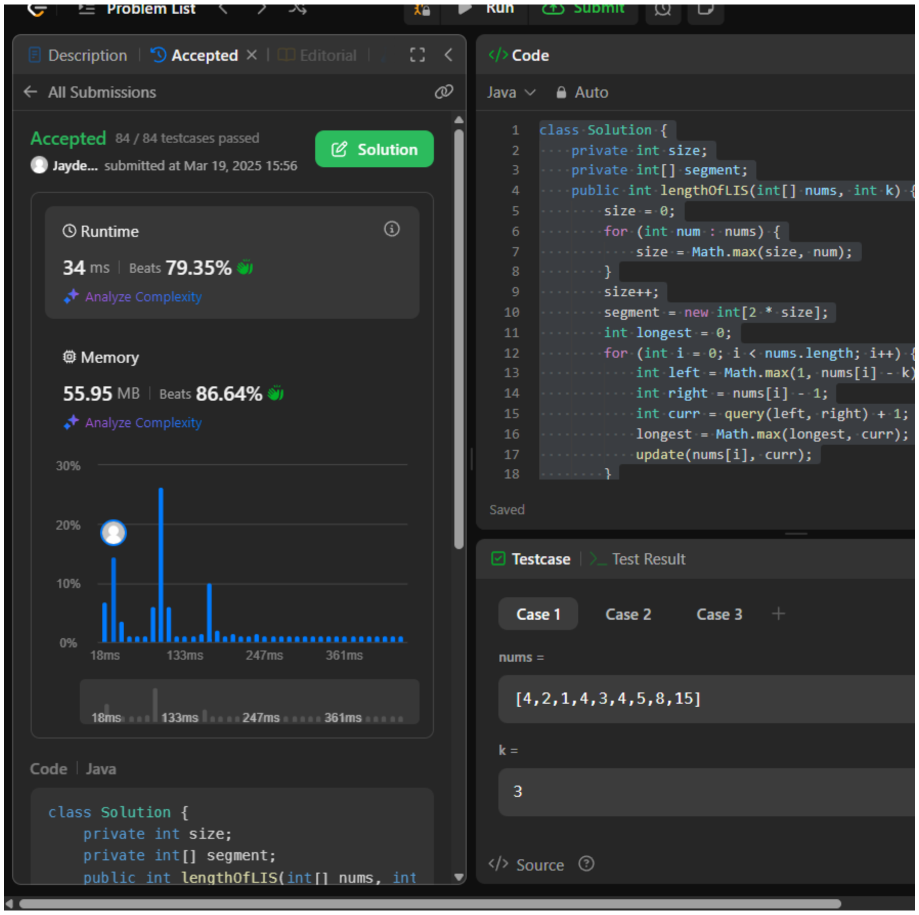
nums =

[1,3,2,3,1]

code:

```
class Solution
{
  int get_pairs(vector<int>& vct , long long int x)
  {
    //sort(vct.begin() , vct.end());
    int size = vct.size();
    int low = 0;
    int high = size - 1;
    int ans = -1;
    while(low <= high)
    {
      int mid = high - (high - low) / 2;
      int ele = vct[mid];
      if(ele > x)
      {
        ans = mid;
        high = mid - 1;
      }
      else
      {
        low = mid  + 1;
      }
    }
    if(ans == -1) return 0;
    return vct.size() - ans;
  }


  // void print_vector(vector<int>& nums)
  // {
  //   cout<<endl;
  //   for(auto it : nums)
  //   {
  //     cout<<" "<<it;
  //   }
  //   cout<<endl;
  // }


public:
  int reversePairs(vector<int>& nums)
  {
    vector<int> vct;
    int counter = 0;
    for(auto it : nums)
    {
      long long int x = 1LL * 2 * it;
      counter += get_pairs(vct , x);
      int low = 0;
      int high = vct.size();
      int ans = vct.size();
      while(low < high)
      {
        int mid = low + (high - low) / 2;
        if(vct[mid] >= it)
        {
          ans = mid;
          high = mid;
        }
        else
        {
          low = mid + 1;
        }
      }
      vct.insert(vct.begin() + ans , it);
      //print_vector(vct);
    }


    return counter;
  }
};
```

question 2:
reverse bits :



All Submissions

Accepted 84 / 84 testcases passed

Jayde... submitted at Mar 19, 2025 15:56

Runtime

34 ms | Beats 79.35%

Memory

55.95 MB | Beats 86.64%

Code | Java

```
class Solution {
    private int size;
    private int[] segment;
    public int lengthOfLIS(int[] nums, int
```

Java Auto

```
class Solution {
    private int size;
    private int[] segment;
    public int lengthOfLIS(int[] nums, int k) {
        size = 0;
        for (int num : nums) {
            size = Math.max(size, num);
        }
        size++;
        segment = new int[2 * size];
        int longest = 0;
        for (int i = 0; i < nums.length; i++) {
            int left = Math.max(1, nums[i] - k)
            int right = nums[i] - 1;
            int curr = query(left, right) + 1;
            longest = Math.max(longest, curr);
            update(nums[i], curr);
        }
```

Saved

Testcase | Test Result

nums =

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

k =

```
3
```

code:

```
class Solution {
  private int size;
  private int[] segment;
  public int lengthOfLIS(int[] nums, int k) {
    size = 0;
    for (int num : nums) {
      size = Math.max(size, num);
    }
    size++;
    segment = new int[2 * size];
    int longest = 0;
    for (int i = 0; i < nums.length; i++) {
      int left = Math.max(1, nums[i] - k);
      int right = nums[i] - 1;
      int curr = query(left, right) + 1;
      longest = Math.max(longest, curr);
      update(nums[i], curr);
    }
    return longest;
  }
  private int query(int left, int right) {
    // edge cases
    if (left > right) {
      return 0;
    }
    // normal cases
    left += size;
    right += size;
    int result = 0;
    while (left <= right) {
      if ((left & 1) == 1) {
        result = Math.max(result, segment[left++]);
      }
      if ((right & 1) == 0) {
        result = Math.max(result, segment[right--]);
      }
      left /= 2;
      right /= 2;
    }
    return result;
  }
  private void update(int index, int value) {
    index += size;
    segment[index] = value;
    for (index /= 2; index >= 1; index /= 2) {
      segment[index] = Math.max(segment[2 * index], segment[2 * index + 1]);
    }
  }
}
```

```
};
```