Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 84 / 84 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 22:12

Solution

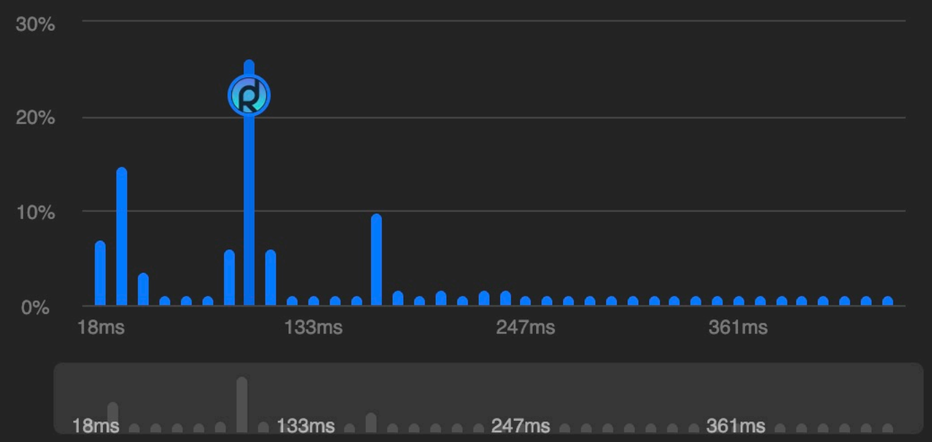
Runtime

106 ms | Beats **39.75%**

Analyze Complexity

Memory

69.08 MB | Beats **5.33%**



Code | Java

```
class Solution {
```

Code

Java | Auto

```
1  class Solution {
2      public int lengthOfLIS(int[] nums, int k) {
3          // Find the maximum value in nums to define the size of the Segment Tree
4          int maxVal = nums[0];
5          for (int value : nums) {
6              maxVal = Math.max(maxVal, value);
7          }
8
9          // Initialize the Segment Tree
10         SegmentTree segmentTree = new SegmentTree(maxVal);
11
12         // Variable to keep track of the length of the Longest Increasing Subsequence (LIS)
13         int maxLength = 0;
14
15         // Iterate through all numbers in the array
```

Saved

Ln 101, Col 2

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

nums =

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

k =

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 140 / 140 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 22:06

Editorial | Solution

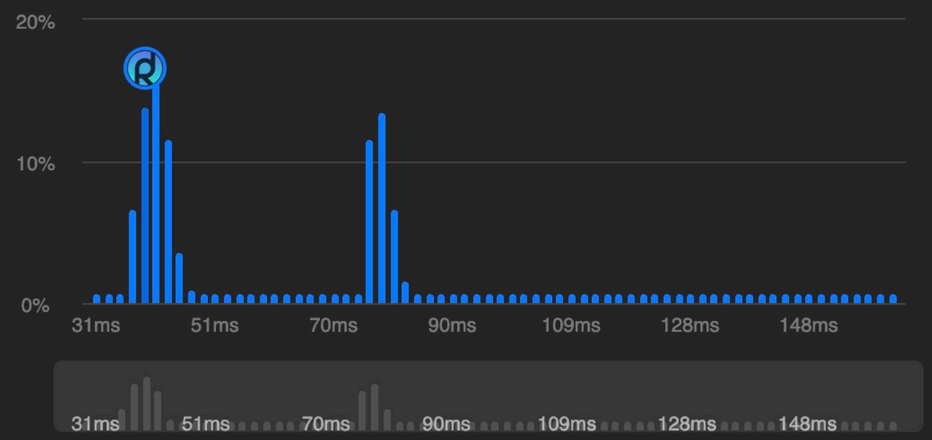
Runtime

40 ms | Beats **92.39%**

Analyze Complexity

Memory

55.06 MB | Beats **55.54%**



Code | Java

```
class Solution {
```

Code

Java | Auto

```
        while (i <= mid && j <= right) {
            if (nums[i] <= nums[j]) {
                temp[k++] = nums[i++];
            } else {
                temp[k++] = nums[j++];
            }
        }

        while (i <= mid) temp[k++] = nums[i++];
        while (j <= right) temp[k++] = nums[j++];

        System.arraycopy(temp, 0, nums, left, temp.length);
    }
}
```

Saved | Ln 42, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

nums =

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

Source

Description | Accepted | Editorial | Solutions | Submis

All Submissions

Accepted 44 / 44 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 22:05

Editorial | Solution

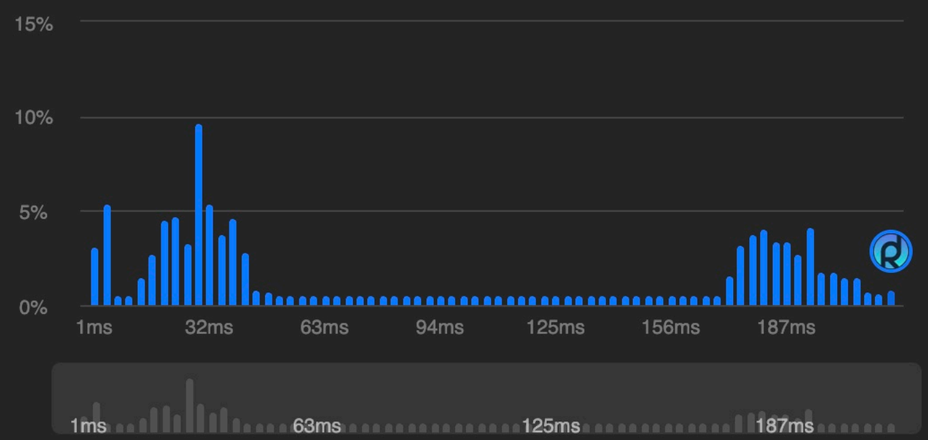
Runtime

241 ms | Beats **18.48%**

Analyze Complexity

Memory

51.35 MB | Beats **61.05%**



Code | Java

```
import java.util.*;
```

Code

Java | Auto

```
            } else {
                heights.remove(point[1]); // Remove building end
            }

            int currentHeight = heights.peek();
            if (currentHeight != prevHeight) {
                result.add(Arrays.asList(point[0], currentHeight));
                prevHeight = currentHeight;
            }
        }

        return result;
    }
}
```

Saved | Ln 38, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

buildings =

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

Source

Description | Accepted | Editorial | Solutions | Submis

All Submissions

Accepted 38 / 38 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 22:03

Editorial | Solution

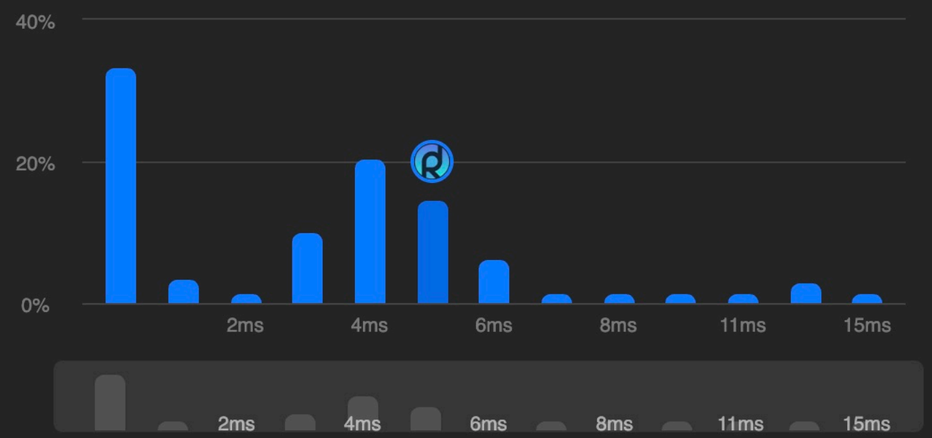
Runtime

5 ms | Beats **31.25%**

Analyze Complexity

Memory

42.29 MB | Beats **84.17%**



Code | Java

```
import java.util.ArrayList;
```

Code

Java | Auto

```
        while (result.size() < n) {
            ArrayList<Integer> temp = new ArrayList<>();
            for (int num : result) {
                if (num * 2 - 1 <= n) temp.add(num * 2 - 1);  // Odd numbers
            }
            for (int num : result) {
                if (num * 2 <= n) temp.add(num * 2);          // Even numbers
            }
            result = temp;
        }

        return result.stream().mapToInt(i -> i).toArray();
    }
}
```

Saved | Ln 22, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

n =

4

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 57 / 57 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 22:02

Solution

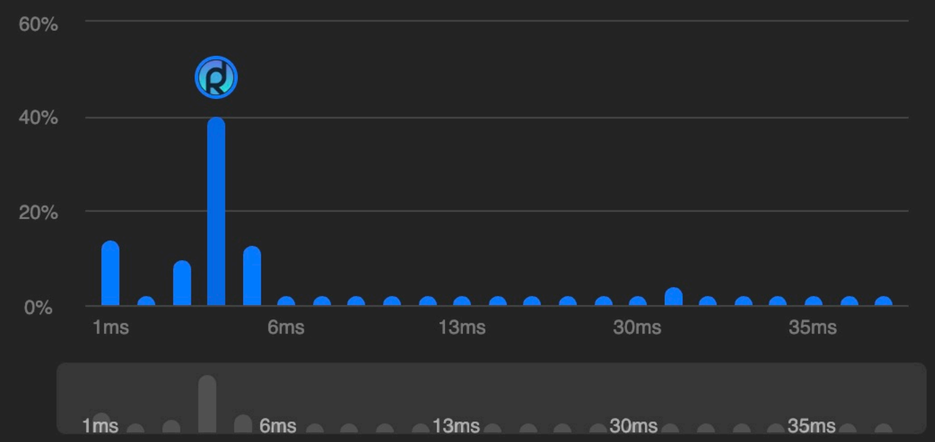
Runtime

4 ms | Beats **74.08%**

Analyze Complexity

Memory

44.56 MB | Beats **37.79%**



Code | Java

```
class Solution {
```

Code

Java | Auto

```
    }

    private int pow(int x, int n) {
        int res = 1;
        while (n > 0) {
            if (n % 2 == 1) {
                res = (res * x) % MOD;
            }
            x = (x * x) % MOD;
            n /= 2;
        }
        return res;
    }
}

```

Saved | Ln 25, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

a =

2

b =

[31

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 130 / 130 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 22:00

Editorial | Solution

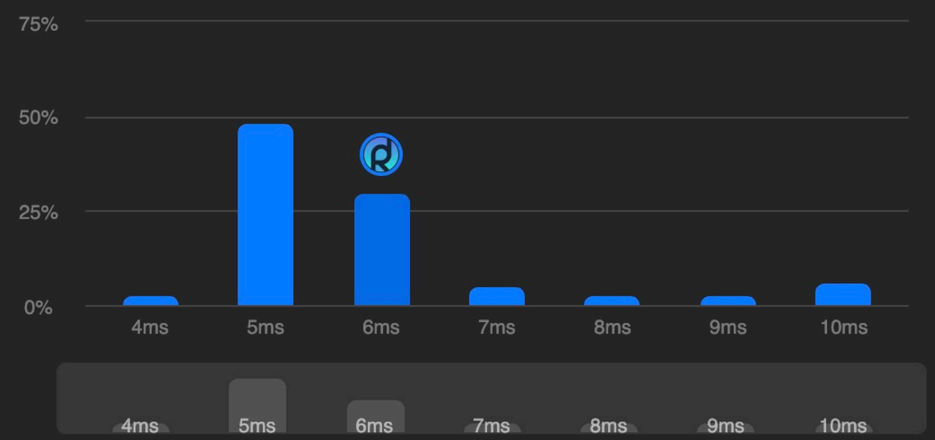
Runtime

6 ms | Beats **51.33%**

Analyze Complexity

Memory

46.29 MB | Beats **20.84%**



Code | Java

```
class Solution {
```

Code

Java | Auto

```
        while (row < matrix.length && col >= 0) {
            if (matrix[row][col] == target) {
                return true;
            } else if (matrix[row][col] > target) {
                col--;  // Move left
            } else {
                row++;  // Move down
            }
        }

        return false;
    }
}
```

Saved | Ln 19, Col 1

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]]
```

target =

```
5
```

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 210 / 210 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 21:59

Editorial | Solution

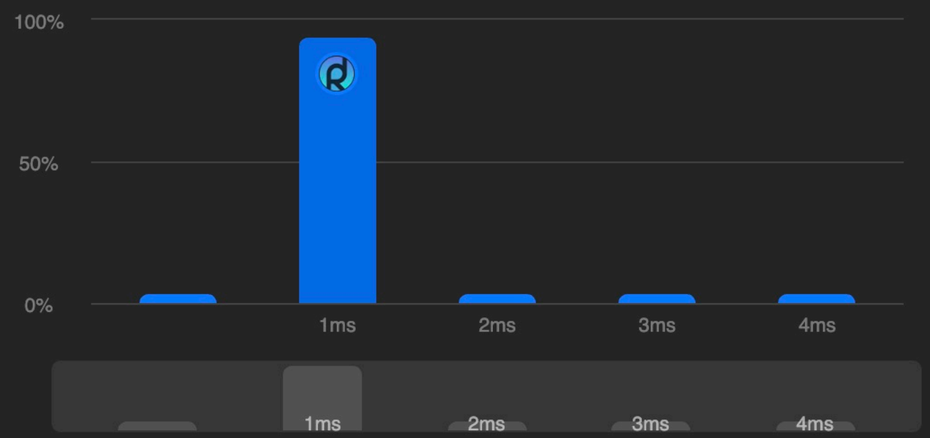
Runtime

1 ms | Beats 99.52%

Analyze Complexity

Memory

57.20 MB | Beats 31.68%



Code | Java

```
class Solution {
```

Code

Java | Auto

```
        int maxSum = nums[0];
        int currentSum = 0;

        for (int num : nums) {
            currentSum += num;
            maxSum = Math.max(maxSum, currentSum);
            if (currentSum < 0) {
                currentSum = 0;
            }
        }

        return maxSum;
    }
}
```

Saved | Ln 17, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

nums =

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

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 598 / 598 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 21:58

Editorial | Solution

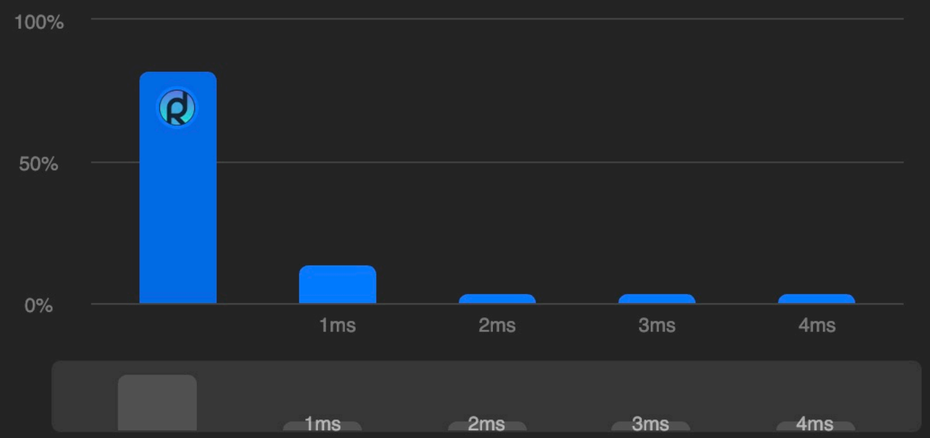
Runtime

0 ms | Beats **100.00%**

Analyze Complexity

Memory

40.52 MB | Beats **85.64%**



Code | Java

```
class Solution {
```

Code

Java | Auto

```
class Solution {
    public int hammingWeight(int n) {
        int count = 0;
        while (n != 0) {
            count += n & 1;  // Add 1 if the last bit is set
            n >>>= 1;        // Unsigned right shift
        }
        return count;
    }
}
```

Saved | Ln 11, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

n =

11

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 600 / 600 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 21:57

Editorial | Solution

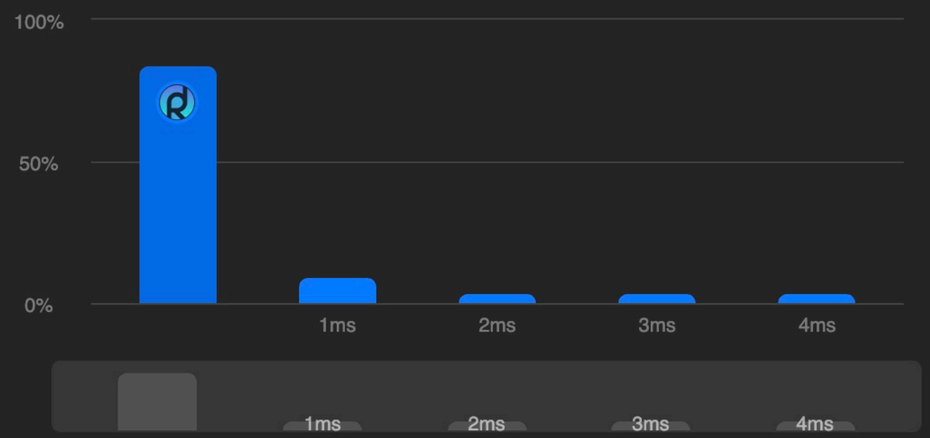
Runtime

0 ms | Beats **100.00%**

Analyze Complexity

Memory

42.34 MB | Beats **9.12%**



Code | Java

```
public class Solution {
```

Code

Java | Auto

```
1  public class Solution {
2      // you need treat n as an unsigned value
3      public int reverseBits(int n) {
4          int result = 0;
5          for (int i = 0; i < 32; i++) {
6              result <<= 1;       // Left shift result to make space
7              result |= (n & 1);  // Add the last bit of n to result
8              n >>>= 1;          // Unsigned right shift n
9          }
10         return result;
11     }
12 }
13
```

Saved | Ln 13, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

n =

```
00000010100101000001111010011100
```

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 73 / 73 testcases passed

jpAYB9fyeP submitted at Mar 18, 2025 21:56

Solution

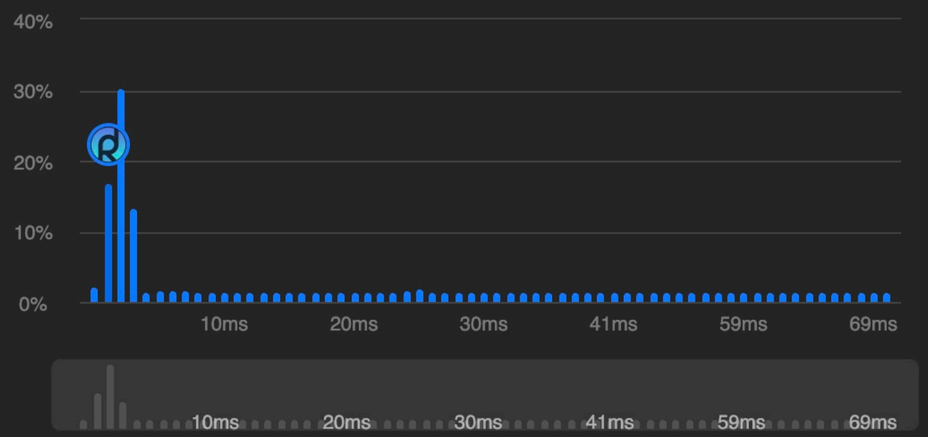
Runtime

1 ms | Beats 97.72%

Analyze Complexity

Memory

41.91 MB | Beats 86.34%



Code | Java

```
class Solution {
```

Code

Java | Auto

```
1  class Solution {
2      public String longestNiceSubstring(String s) {
3          int len = s.length();
4          if (len < 2) return "";
5
6          for (int i = 0; i < len; i++) {
7              char ch = s.charAt(i);
8              if (!s.contains(Character.toString(Character.toUpperCase(ch))) ||
9                  !s.contains(Character.toString(Character.toLowerCase(ch)))) {
10
11                 String leftPart = longestNiceSubstring(s.substring(0, i));
12                 String rightPart = longestNiceSubstring(s.substring(i + 1));
13
14                 if (leftPart.length() >= rightPart.length()) {
15                     return leftPart;
```

Saved

Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

s =

"YazaAay"

Source