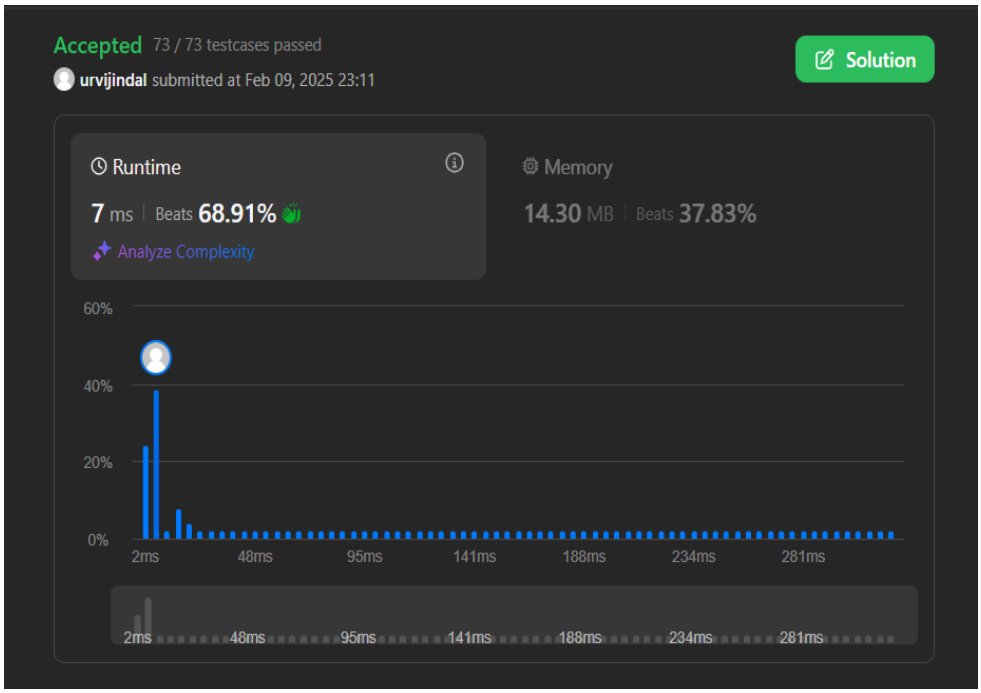
1763.Longest Nice Substring

C++ Auto

```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])) && charSet.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;
    }
};
```



190.[Reverse Bits]

Code

C++ Auto

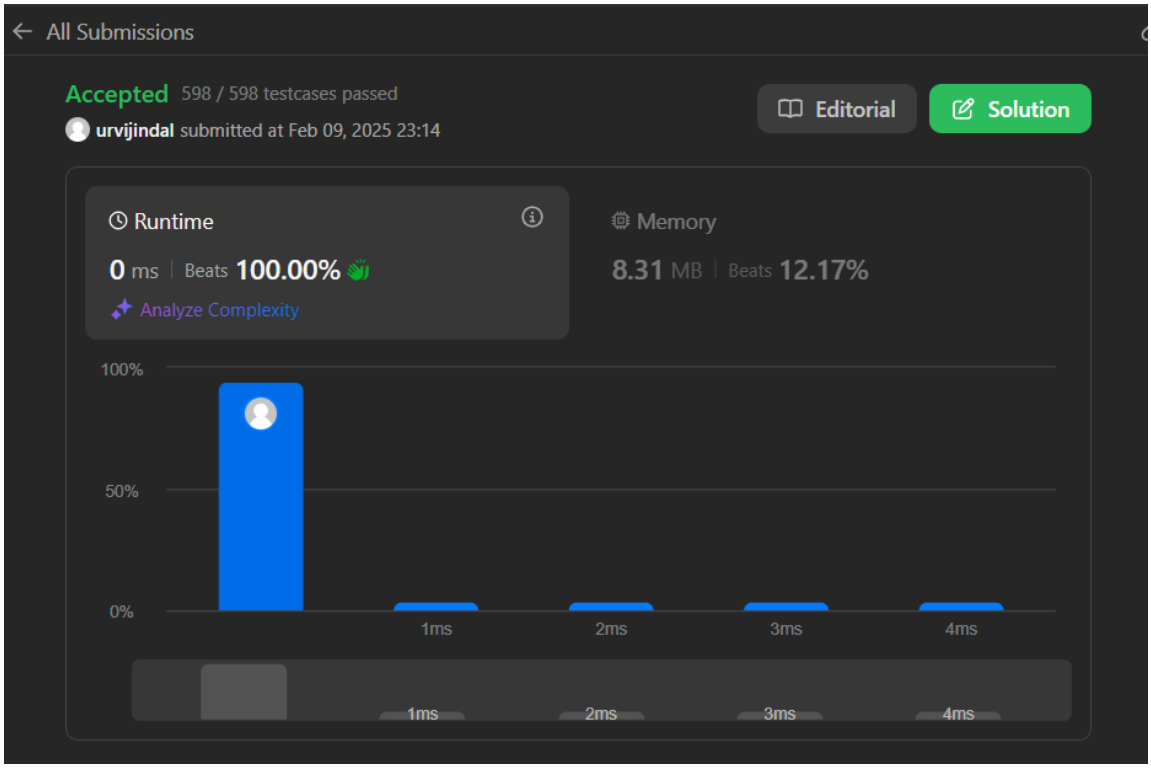
```cpp
class Solution {
public:
    uint32_t reverseBits(uint32_t n) {
        int result=0;
        for(int i=0;i<32;i++)
        {
            result=result<<1;
            result=result|(n&1);
            n=n>>1;
        }
        return result;
    }
};
```

Saved

191.[Number of 1 Bits]

C++ Auto

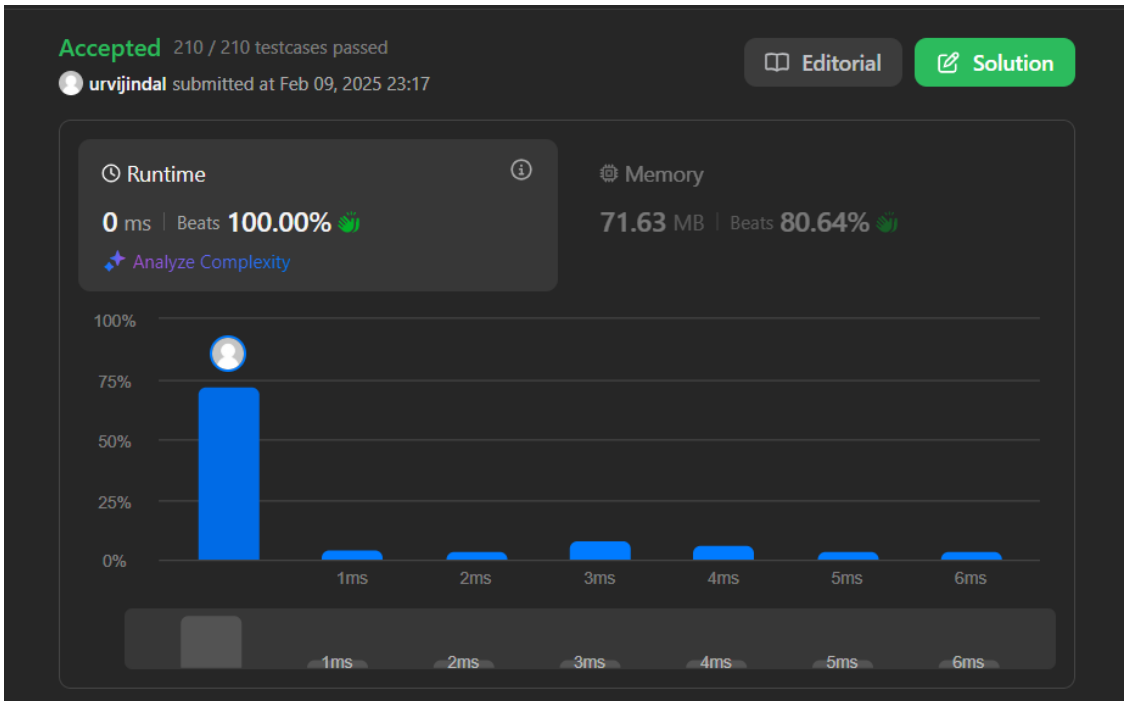
```cpp
class Solution {
public:
    int hammingWeight(int n) {
        int count = 0;
        while (n) {
            count += (n & 1);
            n >>= 1;
        }
        return count;
    }
};
```



53.[Maximum Subarray]

C++ Auto

```cpp
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;
    }
};
```

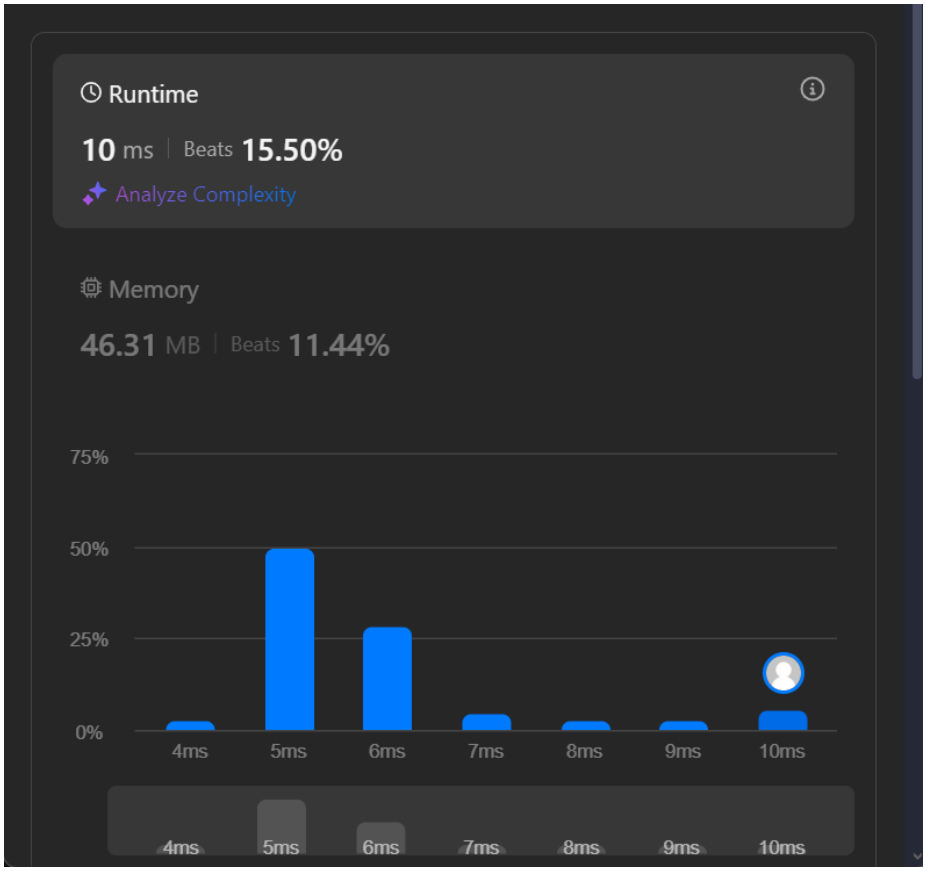


240.[Search a 2D Matrix II]

Code

Java Auto

```java
public class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        for (int i = 0; i < matrix.length; i++) {
            for (int j = 0; j < matrix[i].length; j++) {
                if (matrix[i][j] == target) {
                    return true;
                }
            }
        }
        return false;
    }
}
```

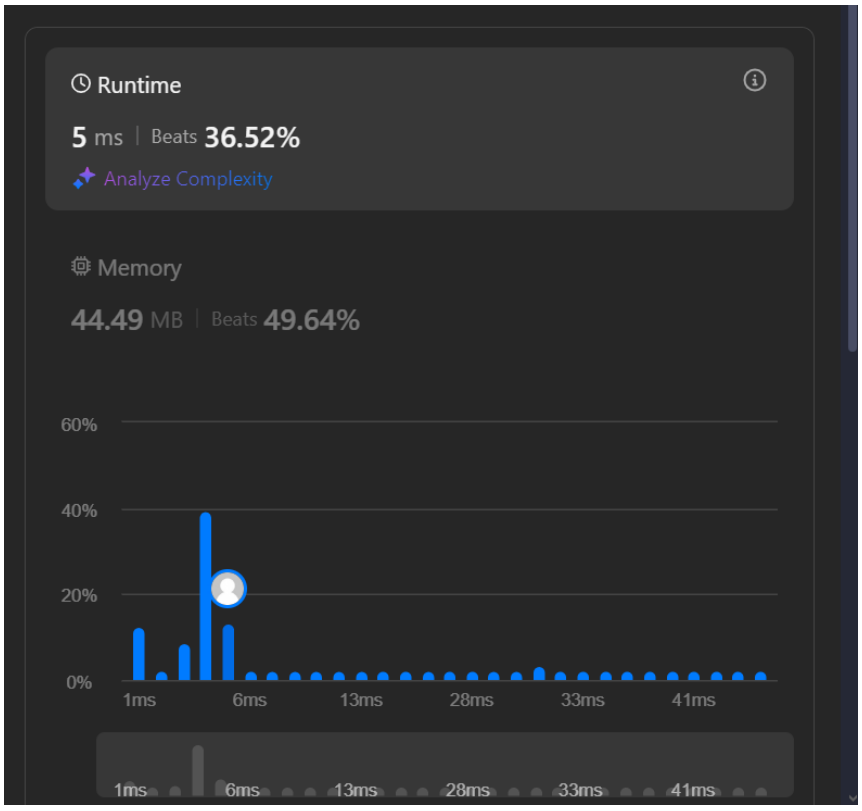


372.Super Pow

Code

Java Auto

```java
class Solution {
  public int superPow(int a, int[] b) {
    int ans = 1;

    a %= kMod;
    for (final int i : b)
      ans = modPow(ans, 10) * modPow(a, i) % kMod;

    return ans;
  }

  private static final int kMod = 1337;

  private int modPow(int x, int n) {
    if (n == 0)
      return 1;
    if (n % 2 == 1)
      return x * modPow(x % kMod, (n - 1)) % kMod;
    return modPow(x * x % kMod, (n / 2)) % kMod;
  }
}
```

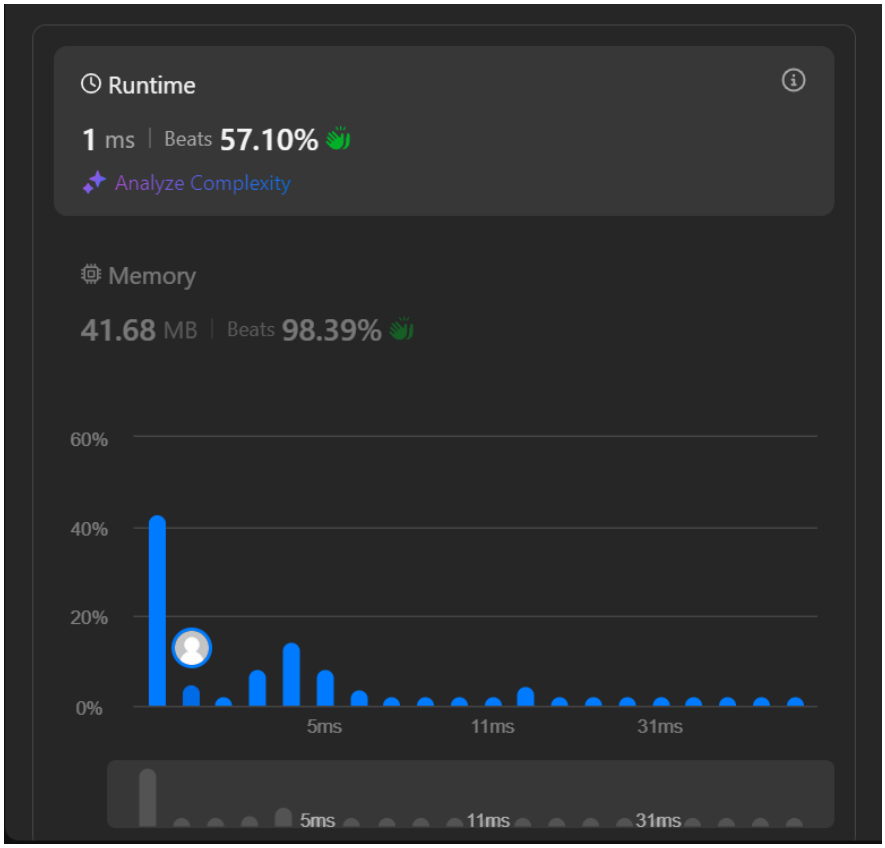


932.[Beautiful Array]

Code

Java Auto

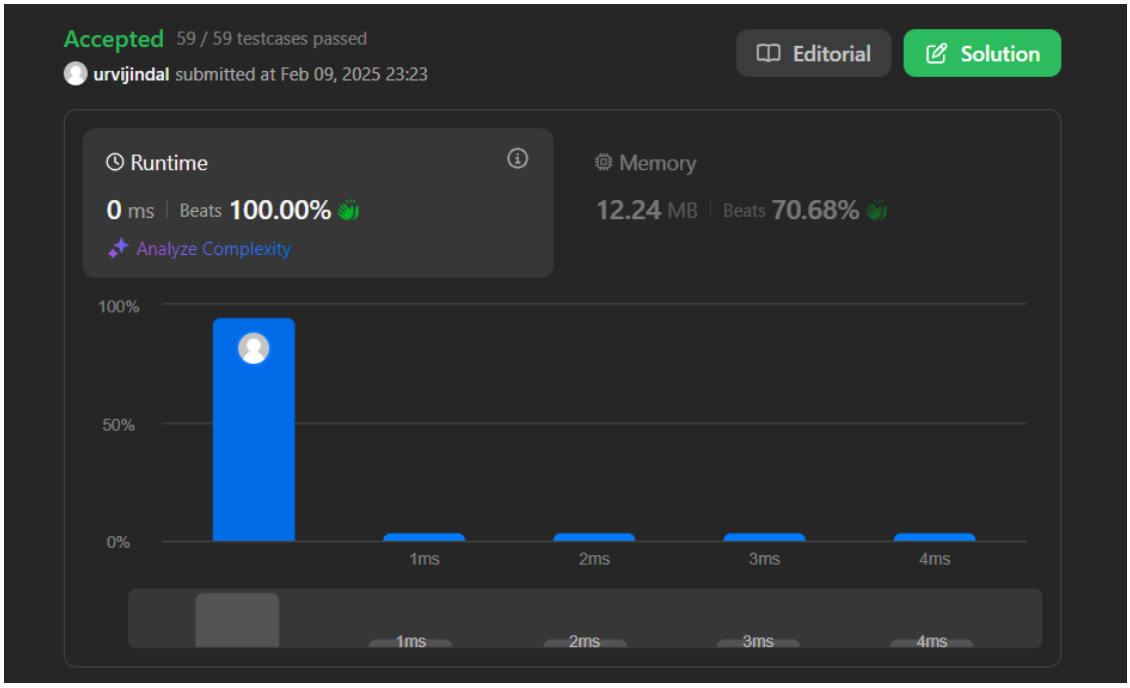
```java
class Solution {
  public int[] beautifulArray(int n) {
    int[] arr = new int[n];
    for (int i = 0; i < n; ++i)
      arr[i] = i + 1;
    divide(arr, 0, n - 1, 1);
    return arr;
  }

  private void divide(int[] arr, int l, int r, int mask) {
    if (l >= r)
      return;
    final int m = partition(arr, l, r, mask);
    divide(arr, l, m, mask << 1);
    divide(arr, m + 1, r, mask << 1);
  }

  private int partition(int[] arr, int l, int r, int mask) {
    int nextSwapped = l;
    for (int i = l; i <= r; ++i)
      if ((arr[i] & mask) > 0)
        swap(arr, i, nextSwapped++);
    return nextSwapped - 1;
  }

  private void swap(int[] arr, int i, int j) {
    final int temp = arr[i];
    arr[i] = arr[j];
```



88.[Merge Sorted Array]

C++ Auto

```cpp
class Solution {
public:
    void merge(vector<int>& nums1, int m, vector<int>& nums2, int n) {
        int i = m - 1, j = n - 1, k = m + n - 1;
        while (i >= 0 && j >= 0) {
            if (nums1[i] > nums2[j]) {
                nums1[k--] = nums1[i--];
            } else {
                nums1[k--] = nums2[j--];
            }
        }
        while (j >= 0) {
            nums1[k--] = nums2[j--];
        }
    }
};
```

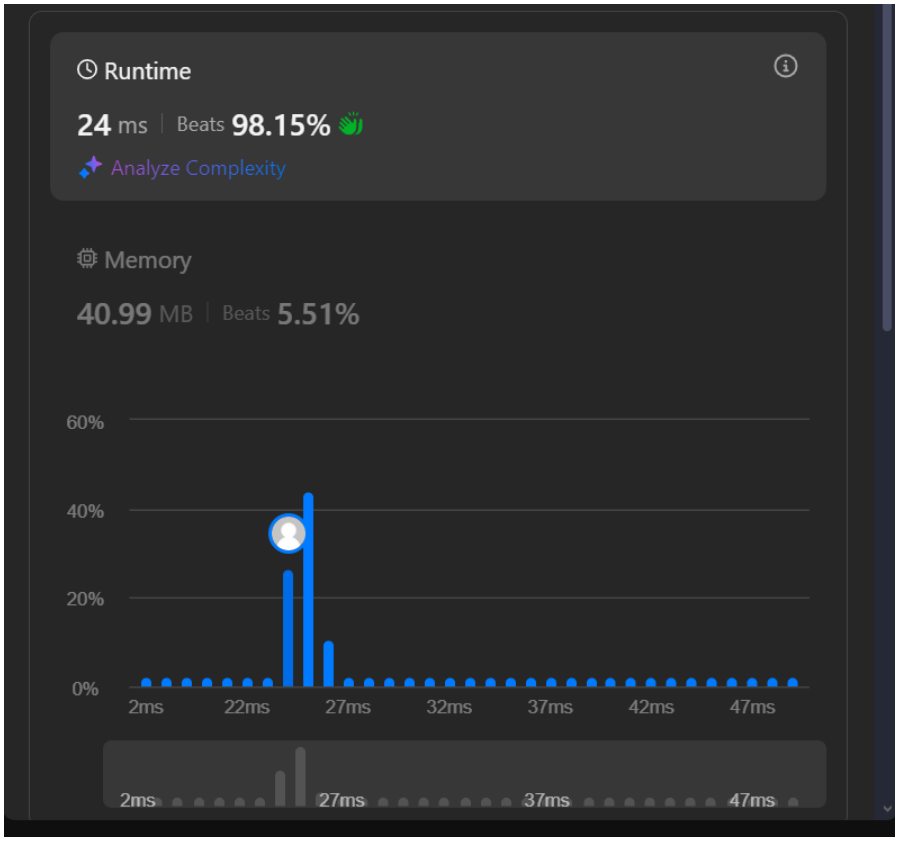


278.First Bad Version

Code

Java Auto

```java
public class Solution extends VersionControl {
  public int firstBadVersion(int n) {
    int l = 1;
    int r = n;

    while (l < r) {
      final int m = l + (r - l) / 2;
      if (isBadVersion(m))
        r = m;
      else
        l = m + 1;
    }

    return l;
  }
}
```



75.Sort Colors

Code

Java Auto

```java
class Solution {
    public void sortColors(int[] nums) {
        HashMap<Integer,Integer> map=new HashMap<>();
        for(int i: nums){
            map.put(i,map.getOrDefault(i,0)+1);
        }
        int x=0;
        for(int i:map.keySet()){
            int y=map.get(i);
            for(int j=0;j<y;j++){
                nums[x++]=i;
            }
        // System.out.println(i);
        }
    }
}
```