

# Problem 1

## Intuition

Given two strings *s* and *t* of lengths *m* and *n* respectively, return the minimum window substring of *s* such that every character in *t* (including duplicates) is included in the window. If there is no such substring, return the empty string "".

## Approach

- 1-First take count of each character in *t* and store them in a map, say hash map
- 2-take two pointers left and right, initially keep left and right at 0
- 3-Make a new Hash Map for current slice in *s* to track count of each character in slice
- 4- Put character at right in Hash Map, then check whether it contains greater or equal number of characters for each character in *t*
- 5-If yes, then store start index of string slice and current Length of slice from that index, then decrease count of character at left in from Hash Map for slice, increase left position by 1
- 6-If no, then increase right index from 1 and again start from step 4
- 7- Repeat above steps unless right is less than *s* length

## Solution

### Code

```
class Solution {  
  
    public String minWindow(String s, String t) {  
  
        Map<Character, Integer> mpForT = new HashMap<>();  
  
        Map<Character, Integer> mpForS = new HashMap<>();  
  
  
        for(char c: t.toCharArray()){  
  
            mpForT.put(c,mpForT.getOrDefault(c, 0) + 1);  
  
        }  
  
        int right = 0, left = 0, minlength = Integer.MAX_VALUE, startInd = -1;  
  
        boolean stringIncreased = true;  
  
  
        while(left <= right && right < s.length()){
```

```

char ch = s.charAt(right);

if(stringIncreased == true)

mpForS.put(ch, mpForS.getOrDefault(ch, 0) + 1);

// System.out.println(mpForS+" -- "+s.substring(left, right + 1));


boolean sliceContainString = true;

for(char tch : mpForT.keySet()){

    if(mpForS.getOrDefault(tch, 0) < mpForT.get(tch)){

        sliceContainString = false;

        break;

    }

}

if(sliceContainString == true){

    // System.out.println(s.substring(left, right + 1));

    // System.out.println(mpForS);

    if((right - left + 1) < minlength){

        startInd = left;

        minlength = right - left + 1;

    }


    char lch = s.charAt(left);

    mpForS.put(lch, mpForS.get(lch) - 1);

    stringIncreased = false;

    left++;

}else{

    stringIncreased = true;

    right++;

}

}

```

```
if(startInd == -1) return "";
```

```
return s.substring(startInd, startInd + minlength);
```

```
}
```

```
}
```

Minimum Window Substring x Abhayraj999999/IOT: This P... x how to use github - Google x Control GitHub via CMD x Git - Downloading Package x +

leetcode.com/problems/minimum-window-substring/submissions/1520871554/

Problem List < > x Run Submit x

</> Code Accepted x

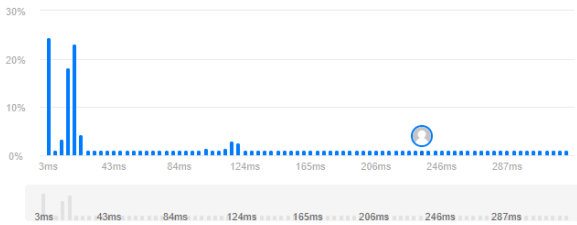
All Submissions

Accepted 268 / 268 testcases passed  
Kanha090 submitted at Jan 26, 2025 13:13

Editorial Solution

Runtime 236 ms | Beats 6.95%  
Memory 45.40 MB | Beats 42.71%

Analyze Complexity



Code | Java

```
class Solution {
```

Activate Windows  
Go to Settings to activate Windows.

Type here to search

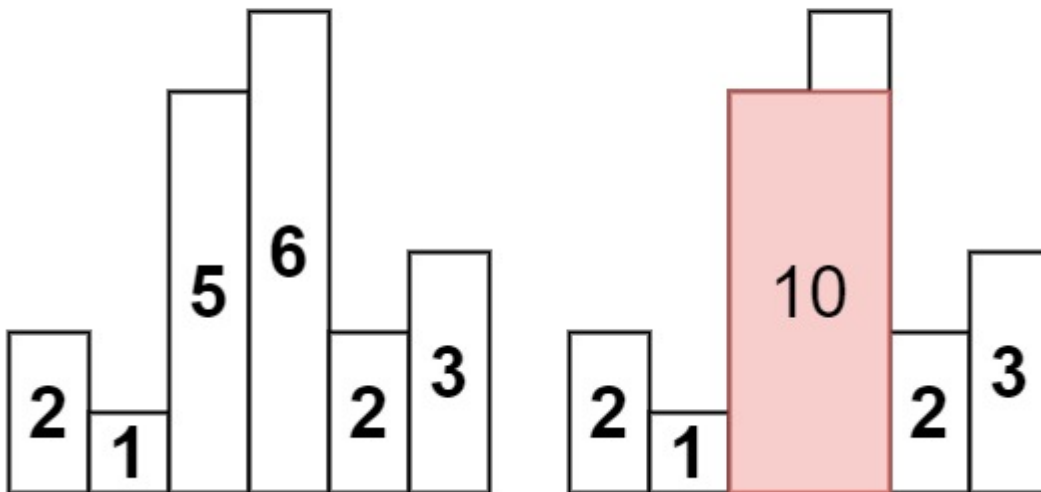
01:18  
27-01-2025

# Problem 2

## Intuition

Given an array of integers heights representing the histogram's bar height where the width of each bar is 1, return the area of the largest rectangle in the histogram.

### Example 1:



## Approach

If we pick a rectangle, say left '2' this time, then think whether we should use it for calculating further area or just left here, the answer is – if next bar is greater than this one then surely we could extend or use it further,

So when we found bar greater than current bar we will use both current and next bar to calculate area,

But if next bar is comparatively smaller than current one then you have to pick the smaller one, and left current bar. Further when you left current bar, you should calculate its area and the area it could further made by adding with its previous bars, for that purpose we will be maintaining stack, that will store the left index of bar which is greater equal to current bar, for calculating area that the current bar will make we will just multiply, width made by bars which are at left and greater equal to current bar from height of current bar, then we compare it with max area till now then update max area, same we do if next bar to be added in our view is less than previous one – here we may use stack data type, but we used list for this, also for the new element is to be added add index of leftmost removed bar as current index for new element, to emphasize the fact that current small bar can merge with these removed bars to make a big rectangle.

After reaching end of array of heights you are left with some values in stack (or list), start removing then one by one, after calculating there area using same method as above and comparing it with current max Area, then updating max Area.

## Solution

### Code

```
class Solution {
```

```
public int largestRectangleArea(int[] heights) {

    List<Integer> values = new ArrayList<>();

    List<Integer> indices = new ArrayList<>();

    int len = heights.length, size;

    int maxArea = 0;

    for(int index = 0; index < len; index++){

        // System.out.println("values = "+values);

        // System.out.println("indices = "+indices);

        int pos = values.size() - 1;

        int newIndexForEle = index;

        while(pos >= 0 && values.get(pos) > heights[index]){

            // System.out.println(index+" -- "+len+" -- "+values.size());

            int area = values.get(pos) * (index - indices.get(pos));

            maxArea = Math.max(area, maxArea);

            newIndexForEle = indices.get(pos);

            values.remove(pos);

            indices.remove(pos);

            pos--;

            // System.out.println("//"+index+" -- "+pos+" -- "+values.size());

        }

        values.add( heights[index] );

        indices.add( newIndexForEle );

    }

    size = values.size();
```

```

while(size > 0){

    maxArea = Math.max(maxArea, values.get(size - 1) * (len - indices.get(size - 1) ));

    values.remove(size - 1);

    indices.remove(size - 1);

    size--;

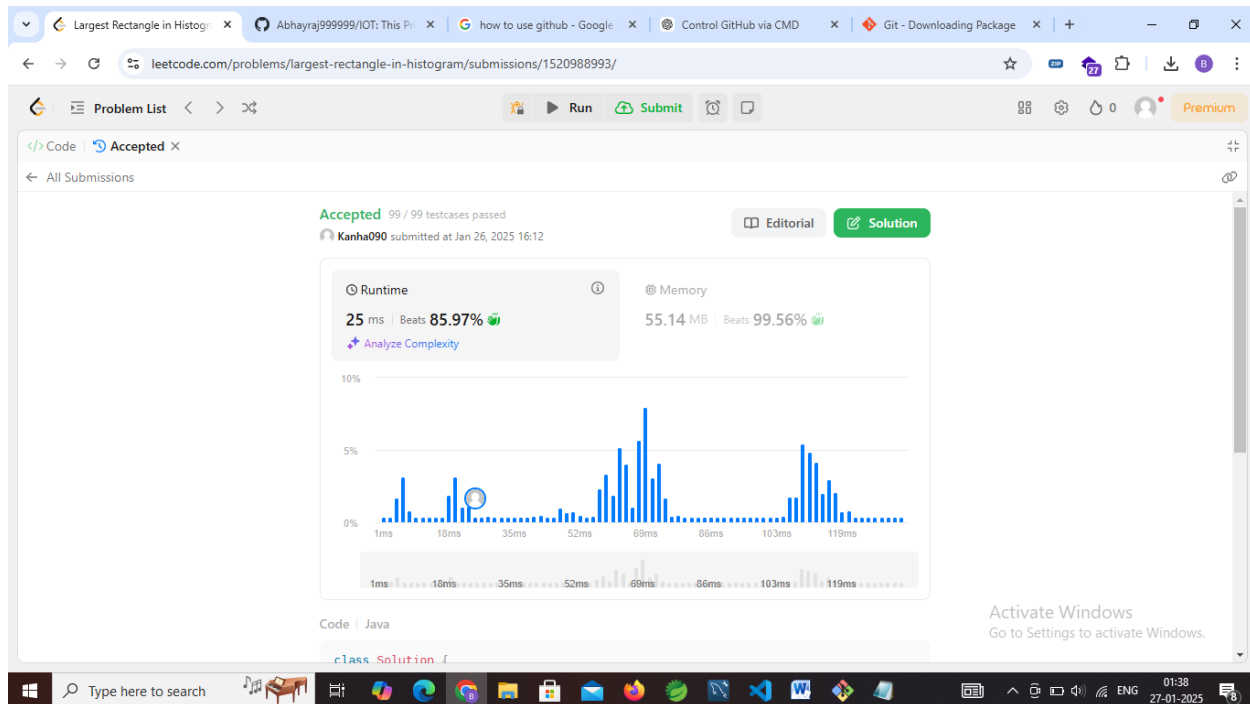
}

return maxArea;

}

}

```



# Problem 3

## Intuition

Given a rows x cols binary matrix filled with 0's and 1's, find the largest rectangle containing only 1's and return its area.

### Example 1:

1	0	1	0	0
1	0	1	1	1
1	1	1	1	1
1	0	0	1	0

## Approach

The approach is quite simple after solving above problem (Problem 2) – we will make histogram starting from each row as base, pass it to previous function then simply compare the max Area from each row, then return the maximum one.

## Solution

### Code

```
class Solution {  
    public int maximalRectangle(char[][] matrix) {  
        int rows = matrix.length;  
        int cols = matrix[0].length;  
        int maxArea = 0;  
        int[][] store = new int[rows][cols];  
  
        for(int r = 0; r < rows; r++){
```

```

for(int c = 0; c < cols; c++){
    if(r == 0){
        store[r][c] = matrix[r][c] == '1' ? 1 : 0;
    }
    else if(matrix[r][c] == '1'){
        store[r][c] = store[r - 1][c] + 1;
    }
}
}

for(int r = 0; r < rows; r++){
    maxArea = Math.max(maxArea, findMaxRect(store[r], cols));
}

```

```

// for(int r = 0; r < rows; r++){
//     for(int c = 0; c < cols; c++){
//         System.out.print(store[r][c]+" ");
//     }
//     System.out.println();
// }

```

```

return maxArea;
}

private int findMaxRect(int[] heights, int len){
    List<Integer> values = new ArrayList<>();
    List<Integer> indices = new ArrayList<>();

    int maxArea = 0;

    for(int index = 0; index < len; index++){
        int pos = values.size() - 1;

```



```
int indexNew = index;

while(pos >= 0 && values.get(pos) > heights[index]){

    int area = values.get(pos) * (index - indices.get(pos));

    maxArea = Math.max(maxArea, area);

    indexNew = indices.get(pos);

    values.remove(pos);

    indices.remove(pos);

    pos--;
}

values.add( heights[index] );

indices.add( indexNew );
}

int size = values.size() - 1;

while(size >= 0){

    int area = values.get(size) * (len - indices.get(size));

    maxArea = Math.max(area, maxArea);

    size--;

}

return maxArea;
}
}
```

Accepted 74 / 74 testcases passed

Kanha090 submitted at Jan 26, 2025 19:04

Runtime: 6 ms | Beats 82.17%

Memory: 45.64 MB | Beats 82.79%

Analyze Complexity

Code | Java

```
class Solution {
```

# Problem 4(DBMS)

## Intuition

Table: Accounts

Column Name	Type
account_id	int
income	int

account\_id is the primary key (column with unique values) for this table.

Each row contains information about the monthly income for one bank account.

Write a solution to calculate the number of bank accounts for each salary category. The salary categories are:

"Low Salary": All the salaries strictly less than \$20000.

"Average Salary": All the salaries in the inclusive range [\$20000, \$50000].

"High Salary": All the salaries strictly greater than \$50000.

The result table must contain all three categories. If there are no accounts in a category, return 0.

Return the result table in any order.

The result format is in the following example.

## Approach

Separately count salaries that comes under each category and union then finally

## Solution

### Code

```
select 'Low Salary' as category, count(income) as accounts_count from Accounts where income < 20000
union
select 'Average Salary' as category, count(income) as accounts_count from Accounts where income >=
20000 and income <= 50000
union
select 'High Salary' as category, count(income) as accounts_count from Accounts where income > 50000
```

SQL 50 - Study Plan - xCount Salary CategorAbhayraj999999/IOT: xhow to use github - xControl GitHub via CIGit - Downloading P

leetcode.com/problems/count-salary-categories/submissions/1521335528/?envType=study-plan-v2&envId=top-sql-50

SQL 50<>SQL 50RunSubmit

</> CodeAccepted xAll Submissions

Accepted11 / 11 testcases passed

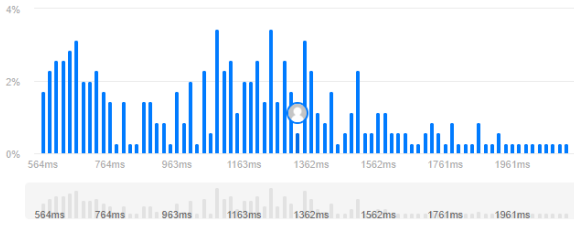
Kanha090 submitted at Jan 26, 2025 23:19

EditorialSolution

Runtime

1336 msBeats 33.24%

Analyze Complexity




564ms764ms963ms1163ms1362ms1562ms1761ms1961ms

CodeOracle

/\* Write your PL/SQL query statement below \*/

Activate Windows  
Go to Settings to activate Windows.

Type here to search



01:56  
27-01-2025

# Problem 5(OOPS)

## Intuition

Create a class "Person" with a static member variable "count" that keeps track of the number of instances created.

## Solution

## Code

```
class Person{

    private String name;

    private static int count;

    Person(String n){

        name = n;

        count++;

    }

    public int getCount(){

        return count;

    }

    public String getName(){

        return name;

    }

    public void setName(String s){

        name = s;

    }

    public String setCount(){

        return "You are not allowed to set count explicitly";

    }

}
```

```
}
```

```
public static void main(String[] args) {
```

```
    Person p1 = new Person("Ram");
```

```
    Person p2 = new Person("Raj");
```

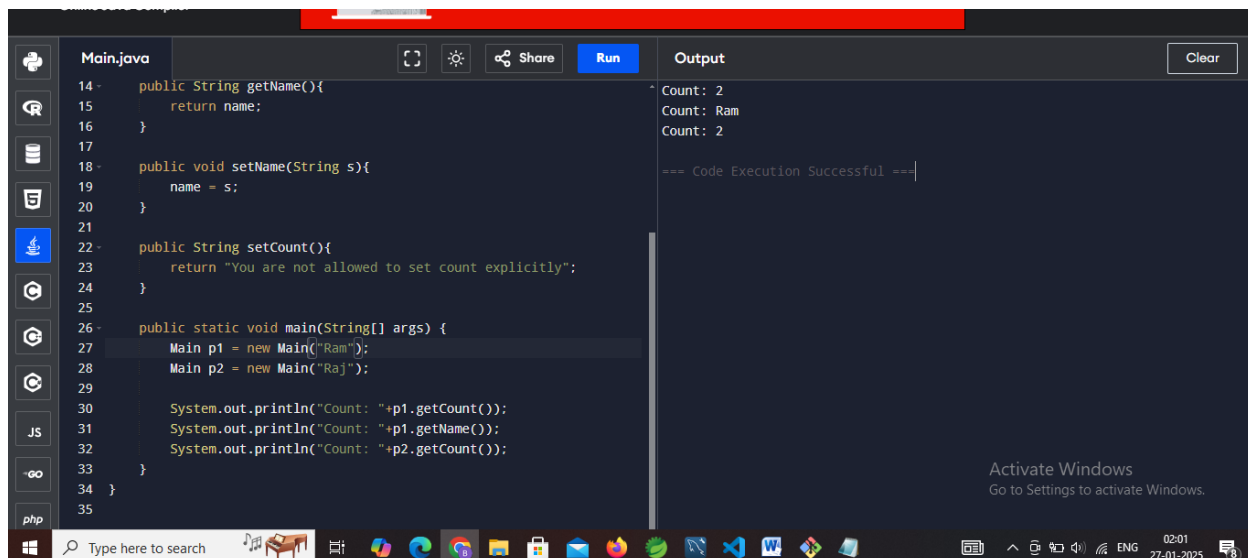
```
    System.out.println("Count: "+p1.getCount());
```

```
    System.out.println("Count: "+p1.getName());
```

```
    System.out.println("Count: "+p2.getCount());
```

```
}
```

```
}
```



The screenshot shows an IDE window with a file named 'Main.java'. The code defines a 'Main' class with methods 'getName()', 'setName()', 'setCount()', and a 'main' method. The 'main' method creates two 'Main' objects, 'p1' (named 'Ram') and 'p2' (named 'Raj'), and prints their 'Count' and 'Name' values. The 'Output' pane on the right shows the execution results: 'Count: 2', 'Count: Ram', and 'Count: 2', followed by a success message. The Windows taskbar at the bottom shows the date as 27-01-2025 and the time as 02:01.

```
Main.java
14 public String getName(){
15     return name;
16 }
17
18 public void setName(String s){
19     name = s;
20 }
21
22 public String setCount(){
23     return "You are not allowed to set count explicitly";
24 }
25
26 public static void main(String[] args) {
27     Main p1 = new Main("Ram");
28     Main p2 = new Main("Raj");
29
30     System.out.println("Count: "+p1.getCount());
31     System.out.println("Count: "+p1.getName());
32     System.out.println("Count: "+p2.getCount());
33 }
34 }
35
```

Output

```
Count: 2
Count: Ram
Count: 2

=== Code Execution Successful ===
```

Activate Windows  
Go to Settings to activate Windows.