AP Assignment

Hard Problems for Fast Learners

Name: Arnav Jain   **UID: 22BCS15161**   **Section: 601-A**

Ques 1. Linked List Cycle https://leetcode.com/problems/linked-list-cycle/description/

Code:

```
class Solution {
public:
    bool hasCycle(ListNode *head) {
        ListNode* slow = head;
        ListNode* fast = head;

        while (fast != nullptr && fast->next != nullptr) {
            slow = slow->next;
            fast = fast->next->next;
            if (slow == fast) {
                return true;
            }
        }

        return false;

    }
};
```

Q2. Longest substring without repeating characters https://leetcode.com/problems/longest-substring-without-repeating-characters/description/

CODE:

```
class Solution {

public:

    int lengthOfLongestSubstring(string s) {

        int left = 0;

        int maxLength = 0;

        unordered_set<char> charSet;


        for (int right = 0; right < s.length(); right++) {

            while (charSet.find(s[right]) != charSet.end()) {

                charSet.erase(s[left]);

                left++;

            }


            charSet.insert(s[right]);
```

```
        maxLength = max(maxLength, right - left + 1);

    }


    return maxLength;

  }

};
```

Q3. Search a 2D matrix II https://leetcode.com/problems/search-a-2d-matrix-ii/description/
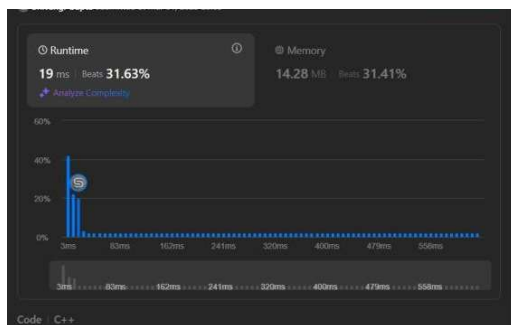
Code:

```
class Solution {

public:

    bool searchMatrix(vector<vector<int>>& matrix, int target) {

        int rows=matrix.size();

        int cols=matrix[0].size();

        if (rows == 0) return false;

        int s=0,e=rows*cols-1;

        while(s<=e){
```

```cpp
        int mid=s+(e-s)/2;

        int midValue=matrix[mid/cols][mid%cols];

        if(target==midValue) return true;

        else if(midValue<target) s=mid+1;

        else e=mid-1;

    }

    return false;

}

};
```
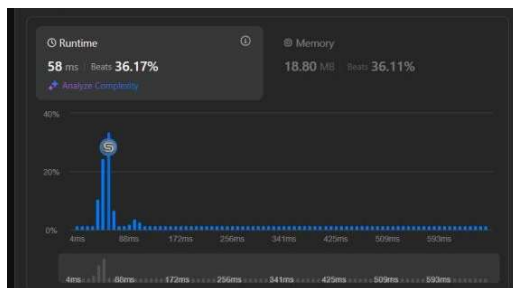
Q4. The Skyline Problem

Code:

```cpp
vector<vector<int>> getSkyline(vector<vector<int>>& buildings) {

    vector<vector<int>> result;

    vector<pair<int, int>> events;


    for (auto& building : buildings) {
```

```cpp
    int left = building[0], right = building[1], height = building[2];

    events.push_back({left, -height});

    events.push_back({right, height});

}


sort(events.begin(), events.end(), [](const pair<int, int>& a, const pair<int, int>& b) {

    if (a.first == b.first) {

        return a.second > b.second;

    }

    return a.first < b.first;

});


priority_queue<int> maxHeap;

maxHeap.push(0);


int prevMaxHeight = 0;


for (auto& event : events) {

    int x = event.first;

    int height = event.second;


    if (height < 0) {

        maxHeap.push(-height);

    } else {

        maxHeap.push(height);

    }


    int currentMaxHeight = maxHeap.top();
```

```
      if (currentMaxHeight != prevMaxHeight) {

        result.push_back({x, currentMaxHeight});

        prevMaxHeight = currentMaxHeight;

      }

    }


    return result;

  }
```
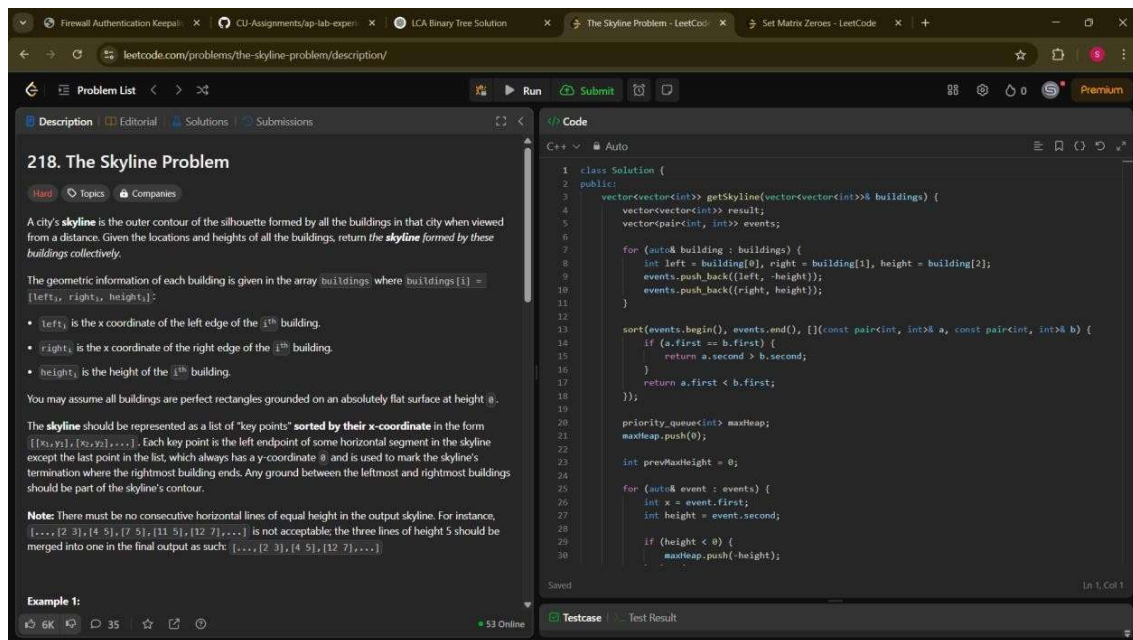

Q5. Set matrix zeroes https://leetcode.com/problems/set-matrix-zeroes/

CODE:

```cpp
class Solution {

public:

  void setZeroes(vector<vector<int>>& matrix) {

    int m = matrix.size(), n = matrix[0].size();

    bool firstColZero = false;
```

```
for (int i = 0; i < m; ++i) {

    if (matrix[i][0] == 0) firstColZero = true;

    for (int j = 1; j < n; ++j) {

        if (matrix[i][j] == 0) {

            matrix[i][0] = 0;

            matrix[0][j] = 0;

        }

    }

}


for (int i = m - 1; i >= 0; --i) {

    for (int j = n - 1; j >= 1; --j) {

        if (matrix[i][0] == 0 || matrix[0][j] == 0)

            matrix[i][j] = 0;

    }

    if (firstColZero)

        matrix[i][0] = 0;

  }

 }

};
```

Firewall Authentication Keepalı ✕ | CU-Assignments/ap-lab-experi ✕ | LCA Binary Tree Solution ✕ | Set Matrix Zeroes - LeetCode ✕ +

leetcode.com/problems/set-matrix-zeroes/

Problem List

Description | Editorial | Solutions | Submissions

73. Set Matrix Zeroes

Medium | Topics | Companies | Hint

Given an `m x n` integer matrix `matrix`, if an element is `0`, set its entire row and column to `0`'s.

You must do it in place.

Example 1:


```
Input: matrix = [[1,1,1],[1,0,1],[1,1,1]]
Output: [[1,0,1],[0,0,0],[1,0,1]]
```

Example 2:

15.5K | 145 | 166 Online

```cpp
class Solution {
public:
    void setZeroes(vector<vector<int>>& matrix) {
        int m = matrix.size(), n = matrix[0].size();
        bool firstColZero = false;

        for (int i = 0; i < m; ++i) {
            if (matrix[i][0] == 0) firstColZero = true;
            for (int j = 1; j < n; ++j) {
                if (matrix[i][j] == 0) {
                    matrix[i][0] = 0;
                    matrix[0][j] = 0;
                }
            }
        }

        for (int i = m - 1; i >= 0; --i) {
            for (int j = n - 1; j >= 1; --j) {
                if (matrix[i][0] == 0 || matrix[0][j] == 0)
                    matrix[i][j] = 0;
            }
            if (firstColZero)
                matrix[i][0] = 0;
        }
    }
};
```

Saved   Ln 24, Col 10

Testcase | Test Result