KWoC Project Report

About KWoC: Kharagpur Winter of Code is a 5 week long online program organized by KOSS for students who are new to open-source development. This program also prepares the student for many open source programmes such as Google Summer of Code, Outreachy, etc.

Project Name: Leet-code

Mentor: Shruti Shreyasi

Project Description: This project aims to solve Leetcode questions to improve the skills of Data Structures and Algorithms.

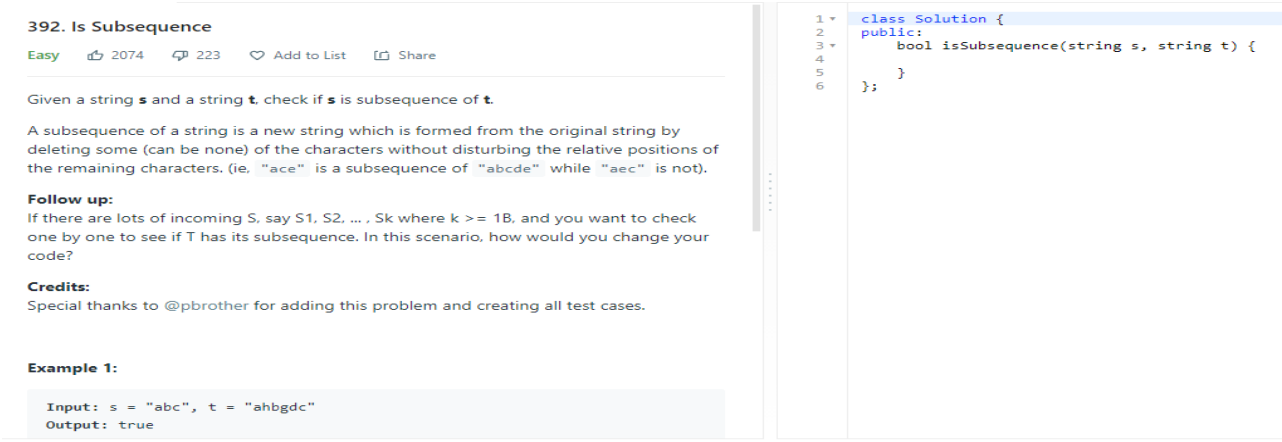
GitHub Repository: https://github.com/shruti170901/Leetcode

My contribution:

I contributed the solutions of three unsolved questions related to binary search in C++. These questions were:

1. Is Subsequence
2. Maximum sum of Rectangle no larger than K
3. Russian Doll Envelopes

Is Subsequence:



392. Is Subsequence

Easy 2074 223 Add to List Share

Given a string **s** and a string **t**, check if **s** is subsequence of **t**.

A subsequence of a string is a new string which is formed from the original string by deleting some (can be none) of the characters without disturbing the relative positions of the remaining characters. (ie, `"ace"` is a subsequence of `"abcde"` while `"aec"` is not).

Follow up:
If there are lots of incoming S, say S1, S2, ... , Sk where k >= 1B, and you want to check one by one to see if T has its subsequence. In this scenario, how would you change your code?

Credits:
Special thanks to @pbrother for adding this problem and creating all test cases.

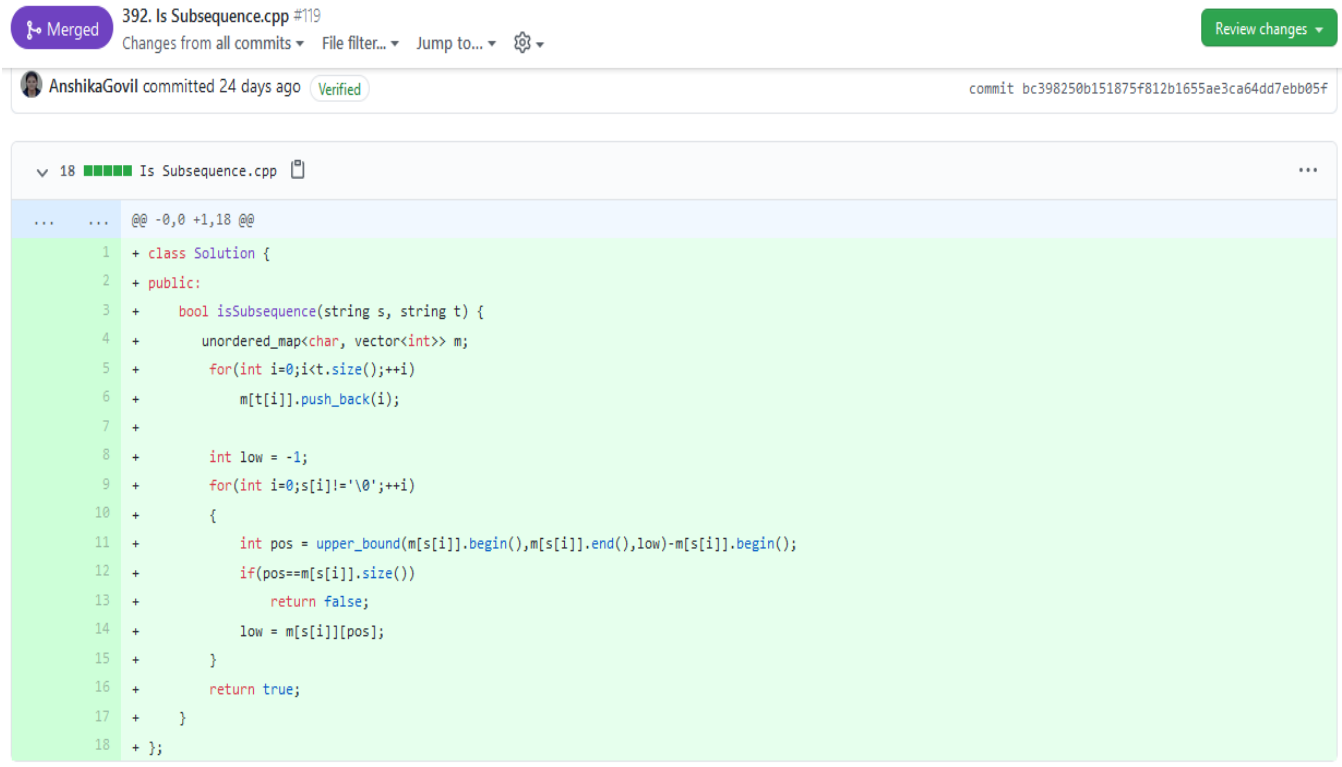
Example 1:

```
Input: s = "abc", t = "ahbgdc"
Output: true
```

```
class Solution {
public:
    bool isSubsequence(string s, string t) {

    }
};
```

Solution:



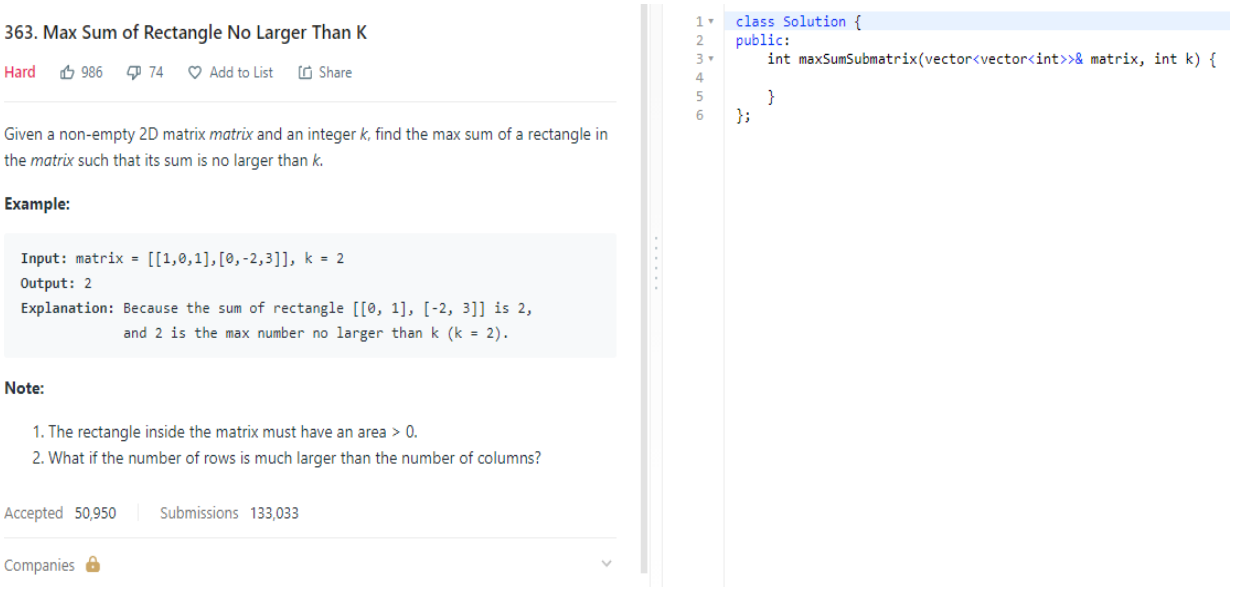
Merged 392. Is Subsequence.cpp #119
Changes from all commits File filter... Jump to... Review changes

AnshikaGovil committed 24 days ago Verified commit bc398250b151875f812b1655ae3ca64dd7ebb05f

18 Is Subsequence.cpp

```
@@ -0,0 +1,18 @@
+ class Solution {
+ public:
+     bool isSubsequence(string s, string t) {
+        unordered_map<char, vector<int>> m;
+         for(int i=0;i<t.size();++i)
+             m[t[i]].push_back(i);
+
+         int low = -1;
+         for(int i=0;s[i]!='\0';++i)
+         {
+             int pos = upper_bound(m[s[i]].begin(),m[s[i]].end(),low)-m[s[i]].begin();
+             if(pos==m[s[i]].size())
+                 return false;
+             low = m[s[i]][pos];
+         }
+         return true;
+     }
+ };
```

Maximum sum of Rectangle no larger than K:



363. Max Sum of Rectangle No Larger Than K

Hard 986 74 Add to List Share

Given a non-empty 2D matrix *matrix* and an integer *k*, find the max sum of a rectangle in the *matrix* such that its sum is no larger than *k*.

Example:

```
Input: matrix = [[1,0,1],[0,-2,3]], k = 2
Output: 2
Explanation: Because the sum of rectangle [[0, 1], [-2, 3]] is 2,
             and 2 is the max number no larger than k (k = 2).
```

Note:

1. The rectangle inside the matrix must have an area > 0.
2. What if the number of rows is much larger than the number of columns?

Accepted 50,950 Submissions 133,033

Companies

```
class Solution {
public:
    int maxSumSubmatrix(vector<vector<int>>& matrix, int k) {

    }
};
```

Solution:

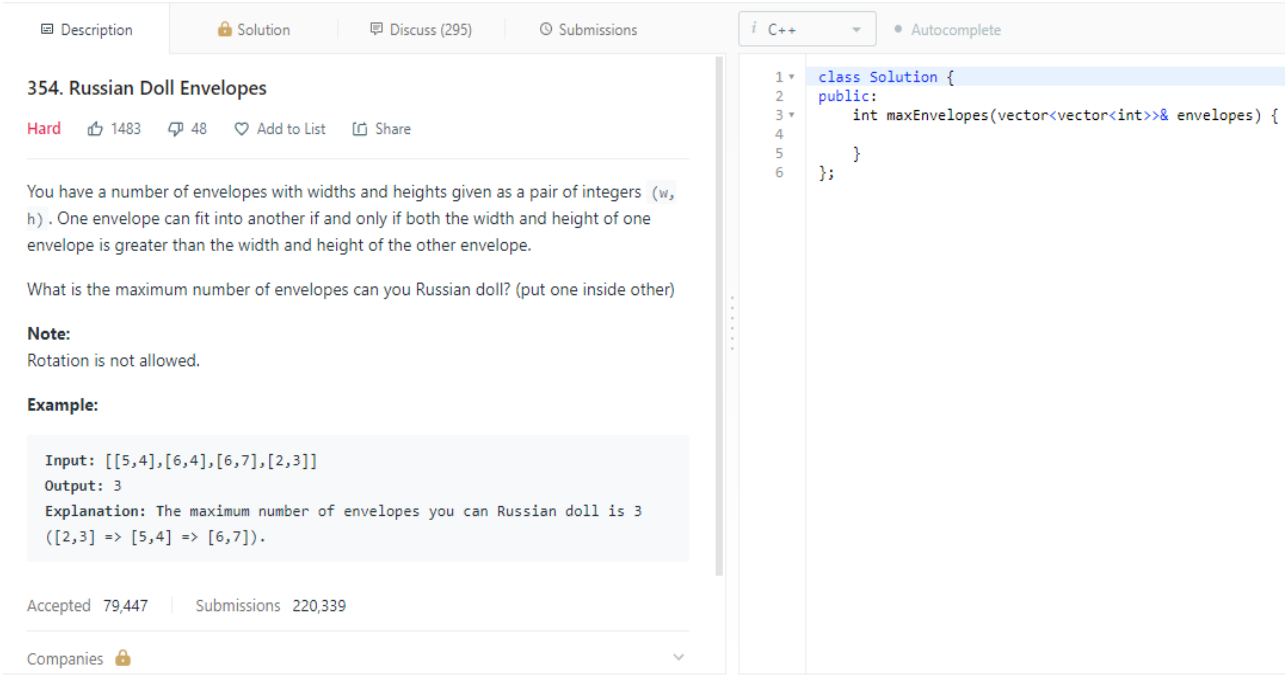
Merged solutions related to binary search #173

Changes from 1 commit File filter... Jump to... Review changes

26 363. Max Sum of Rectangle No Larger Than K.cpp

```
+ class Solution {
+ public:
+     int maxSumSubmatrix(vector<vector<int>>& matrix, int k) {
+       int n = matrix.size();
+       int m = matrix[0].size();
+       int ans = INT_MIN;
+       for(int l = 0; l < m; l++){
+          vector <int> rowSum(n);
+          for(int r = l; r < m; r++){
+             for(int i = 0; i < n; i++)rowSum[i] += matrix[i][r];
+             set < int > s;
+             s.insert(0);
+             int currSum = 0;
+             for(int i = 0; i < n; i++){
+                currSum += rowSum[i];
+                set <int> :: iterator it = s.lower_bound(currSum - k);
+                if(it != s.end()){
+                   ans = max(ans, (currSum - *it));
+                }
+                s.insert(currSum);
+             }
+          }
+       }
+       return ans;
+     }
```

Russian Doll Envelopes:



Description | Solution | Discuss (295) | Submissions | C++ | Autocomplete

354. Russian Doll Envelopes

Hard 1483 48 Add to List Share

You have a number of envelopes with widths and heights given as a pair of integers `(w, h)`. One envelope can fit into another if and only if both the width and height of one envelope is greater than the width and height of the other envelope.

What is the maximum number of envelopes can you Russian doll? (put one inside other)

Note:
Rotation is not allowed.

Example:

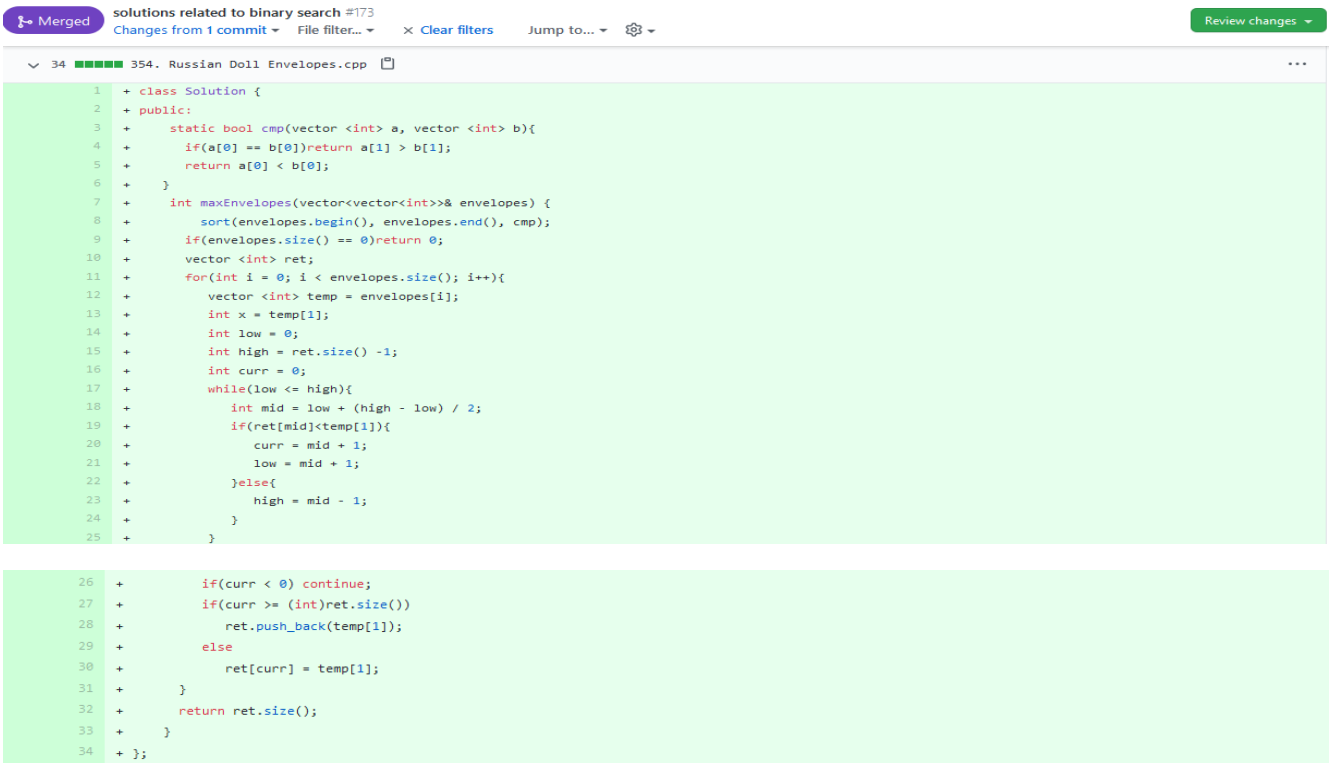
```
Input: [[5,4],[6,4],[6,7],[2,3]]
Output: 3
Explanation: The maximum number of envelopes you can Russian doll is 3
([2,3] => [5,4] => [6,7]).
```

Accepted 79,447 Submissions 220,339

Companies

```
1  class Solution {
2  public:
3      int maxEnvelopes(vector<vector<int>>& envelopes) {
4
5      }
6  };
```

Solution:



Merged solutions related to binary search #173
Changes from 1 commit File filter... Clear filters Jump to... Review changes

34 354. Russian Doll Envelopes.cpp

```
+ class Solution {
+ public:
+     static bool cmp(vector <int> a, vector <int> b){
+       if(a[0] == b[0])return a[1] > b[1];
+       return a[0] < b[0];
+    }
+     int maxEnvelopes(vector<vector<int>>& envelopes) {
+         sort(envelopes.begin(), envelopes.end(), cmp);
+       if(envelopes.size() == 0)return 0;
+       vector <int> ret;
+       for(int i = 0; i < envelopes.size(); i++){
+          vector <int> temp = envelopes[i];
+          int x = temp[1];
+          int low = 0;
+          int high = ret.size() -1;
+          int curr = 0;
+          while(low <= high){
+             int mid = low + (high - low) / 2;
+             if(ret[mid]<temp[1]){
+                curr = mid + 1;
+                low = mid + 1;
+             }else{
+                high = mid - 1;
+             }
+          }
+          if(curr < 0) continue;
+          if(curr >= (int)ret.size())
+             ret.push_back(temp[1]);
+          else
+             ret[curr] = temp[1];
+       }
+       return ret.size();
+     }
+ };
```

Verdict:

I am thankful to Shruti Shreyasi for her guidance in the project. I am also thankful to KOSS, IIT Kharagpur for organizing this beginner friendly program. It helped me to learn open-source contribution.

- *Anshika Govil*