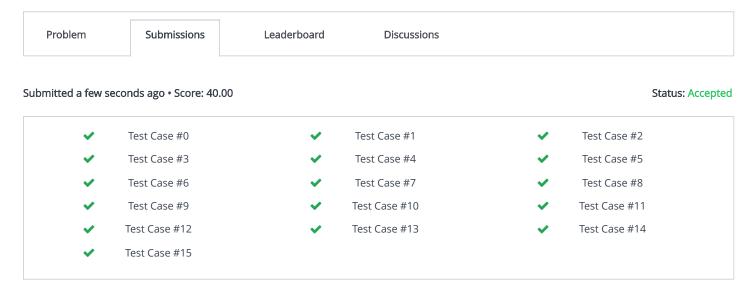
All Contests > In21-CS2023-Lab3 > Array Manipulation

Array Manipulation



Submitted Code

```
Open in editor
  nguage: C++
 1 #include <bits/stdc++.h>
 3 using namespace std;
 5 string ltrim(const string &);
 6 string rtrim(const string &);
 7 vector<string> split(const string &);
 8
9 /*
10 * Complete the 'arrayManipulation' function below.
11 *
12 * The function is expected to return a LONG_INTEGER.
13 * The function accepts following parameters:
   * 1. INTEGER n
14
15
   * 2. 2D_INTEGER_ARRAY queries
   */
16
17
18 long arrayManipulation(int n, vector<vector<int>> queries) {
19
20
      int no_of_operations = queries.size(); // finding the number of operations using vector size
21
      vector<long> final_array(n); // initializing final array
22
23
      for (int i=0; i<no_of_operations; i++){ // looping no of operations times
           int start = queries[i][0];
24
25
           int end = queries[i][1];
           int adder = queries[i][2];
26
27
28
           // using cumilative some method to reduce time complexity
```

```
29
           final_array[start-1] += adder;
30
           if (end < n){
               final_array[end] -= adder;
31
           }
32
       }
33
34
35
       long max = 0;
36
       long sum = 0;
37
       // finding the maximum value
       for (int i=0; i<n; i++){
38
           sum+=final_array[i];
39
40
           if (sum > max){
41
               max = sum;
42
43
       }
       return max;
44
45
46 // we can also use nested for loops but they'll increase time complexity as the input size becomes
  larger
47 //
          int no_of_operations = queries.size();
          vector<long> final_array(n,0);
48 //
49
          for (int i=0; i<no_of_operations; i++){</pre>
50 //
51 //
              for (int j=queries[i][0]; j<=queries[i][1]; j++){</pre>
52 //
                   final_array[j-1] += queries[i][2];
53 //
              }
54 //
          }
55
56 //
          long max = final_array[0];
57 //
          for (int i=0; i<n; i++){
58 //
              if(final_array[i]>max){
59 //
                  max = final_array[i];
60 //
61 //
          }
62
63 }
64
65 int main()
66 {
67
       ofstream fout(getenv("OUTPUT_PATH"));
68
       string first_multiple_input_temp;
69
       getline(cin, first_multiple_input_temp);
70
71
       vector<string> first_multiple_input = split(rtrim(first_multiple_input_temp));
72
73
74
       int n = stoi(first_multiple_input[0]);
75
76
       int m = stoi(first_multiple_input[1]);
77
78
       vector<vector<int>> queries(m);
79
       for (int i = 0; i < m; i++) {
80
           queries[i].resize(3);
81
82
83
           string queries_row_temp_temp;
           getline(cin, queries_row_temp_temp);
84
85
86
           vector<string> queries_row_temp = split(rtrim(queries_row_temp_temp));
87
88
           for (int j = 0; j < 3; j++) {
89
               int queries_row_item = stoi(queries_row_temp[j]);
90
91
               queries[i][j] = queries_row_item;
92
           }
       }
93
```

```
94
95
        long result = arrayManipulation(n, queries);
96
97
        fout << result << "\n";</pre>
98
99
        fout.close();
100
        return 0;
101
102 }
103
104 string ltrim(const string &str) {
        string s(str);
105
106
        s.erase(
107
108
            s.begin(),
            find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
109
        );
110
111
112
        return s;
113 }
114
115 string rtrim(const string &str) {
116
        string s(str);
117
118
        s.erase(
            find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
119
120
            s.end()
121
        );
122
123
        return s;
124 }
125
126 vector<string> split(const string &str) {
127
        vector<string> tokens;
128
129
        string::size_type start = 0;
        string::size_type end = 0;
130
131
        while ((end = str.find(" ", start)) != string::npos) {
132
            tokens.push_back(str.substr(start, end - start));
133
134
            start = end + 1;
135
        }
136
137
        tokens.push_back(str.substr(start));
138
139
140
        return tokens;
141 }
142
```

Interview Prep | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy |