

All Contests > In21-CS2023-Lab5 > Lily's Homework

Lily's Homework

locked

Problem	Submissions	Leaderboard	Discussions
---------	-------------	-------------	-------------

Submitted 2 minutes ago • Score: 35.00

Status: Accepted

✓	Test Case #0	✓	Test Case #1	✓	Test Case #2
✓	Test Case #3	✓	Success 0.02s	✓	Test Case #5
✓	Test Case #6	✓	Download	✓	Test Case #8
✓	Test Case #9	✓	Test Case #10	✓	Test Case #11

Submitted Code

Language: C++

Open in editor

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 string ltrim(const string &);
6 string rtrim(const string &);
7 vector<string> split(const string &);
8
9 /*
10  * Complete the 'lilysHomework' function below.
11  *
12  * The function is expected to return an INTEGER.
13  * The function accepts INTEGER_ARRAY arr as parameter.
14  */
15
16 int lilysHomework(vector<int> arr) {
17     vector<int> arr1=arr, ascArr=arr, descArr=arr;
18     unordered_map<int,int>ascending,descending;
19     int n = (int)arr.size();
20
21     for(int i=0;i<n;i++)
22         ascending[arr[i]]=i;
23     for(int i=0;i<n;i++)
24         descending[arr1[i]]=i;
25
26     // doing quicksort to ascending order
27     int end = n-1;
28     int start = 0;
29     int stack[end - start + 1];
30     int top = -1;
31     stack[++top] = start;
32     stack[++top] = end;
```

```
33
34 while (top >= 0)
35 {
36     end = stack[top--];
37     start = stack[top--];
38
39     int i = start, j = end;
40     int tmp;
41     int pivot = ascArr[(start + end) / 2];
42
43     /* partition */
44     while (i <= j)
45     {
46         while (ascArr[i] < pivot)
47             i++;
48         while (ascArr[j] > pivot)
49             j--;
50         if (i <= j)
51         {
52             tmp = ascArr[i];
53             ascArr[i] = ascArr[j];
54             ascArr[j] = tmp;
55             i++;
56             j--;
57         }
58     };
59
60     /* push values to stack */
61     if (i < end)
62     {
63         stack[++top] = i;
64         stack[++top] = end;
65     }
66     if (start < j)
67     {
68         stack[++top] = start;
69         stack[++top] = j;
70     }
71 }
72 //-----
73
74 // doing quicksort in descending order
75 int end2 = n-1;
76 int start2 = 0;
77 int stack2[end2 - start2 + 1];
78 int top2 = -1;
79 stack2[++top2] = start2;
80 stack2[++top2] = end2;
81
82 while (top2 >= 0)
83 {
84     end2 = stack2[top2--];
85     start2 = stack2[top2--];
86
87     int i = start2, j = end2;
88     int tmp;
89     int pivot = descArr[(start2 + end2) / 2];
90
91     /* partition */
92     while (i <= j)
93     {
94         while (descArr[i] > pivot)
95             i++;
96         while (descArr[j] < pivot)
97             j--;
98         if (i <= j)
```

```
99         {
100             tmp = descArr[i];
101             descArr[i] = descArr[j];
102             descArr[j] = tmp;
103             i++;
104             j--;
105         }
106     };
107
108     /* push values to stack */
109     if (i < end2)
110     {
111         stack2[++top2] = i;
112         stack2[++top2] = end2;//
113     }
114     if (start2 < j)
115     {
116         stack2[++top2] = start2;
117         stack2[++top2] = j;
118     }
119 }
120 // -----
121
122 int swapsAsc=0,swapsDesc=0;
123 for(int i=0;i<n;i++){
124     if(arr[i]!=ascArr[i]){
125         swapsAsc++;
126         int temp=ascending[ascArr[i]];
127         ascending[arr[i]]=temp;
128         swap(arr[i],arr[temp]);
129     }
130 }
131 for(int i=0;i<n;i++){
132     if(arr1[i]!=descArr[i]){
133         swapsDesc++;
134         int temp=descending[descArr[i]];
135         descending[arr1[i]]=temp;
136         swap(arr1[i],arr1[temp]);
137     }
138 }
139 return min(swapsAsc,swapsDesc);
140 }
141
142 int main()
143 {
144     ofstream fout(getenv("OUTPUT_PATH"));
145
146     string n_temp;
147     getline(cin, n_temp);
148
149     int n = stoi(ltrim(rtrim(n_temp)));
150
151     string arr_temp_temp;
152     getline(cin, arr_temp_temp);
153
154     vector<string> arr_temp = split(rtrim(arr_temp_temp));
155
156     vector<int> arr(n);
157
158     for (int i = 0; i < n; i++) {
159         int arr_item = stoi(arr_temp[i]);
160
161         arr[i] = arr_item;
162     }
163
164     int result = lilysHomework(arr);
```

```
165
166     fout << result << "\n";
167
168     fout.close();
169
170     return 0;
171 }
172
173 string ltrim(const string &str) {
174     string s(str);
175
176     s.erase(
177         s.begin(),
178         find_if(s.begin(), s.end(), not1(ptr_fun<int, int>(isspace)))
179     );
180
181     return s;
182 }
183
184 string rtrim(const string &str) {
185     string s(str);
186
187     s.erase(
188         find_if(s.rbegin(), s.rend(), not1(ptr_fun<int, int>(isspace))).base(),
189         s.end()
190     );
191
192     return s;
193 }
194
195 vector<string> split(const string &str) {
196     vector<string> tokens;
197
198     string::size_type start = 0;
199     string::size_type end = 0;
200
201     while ((end = str.find(" ", start)) != string::npos) {
202         tokens.push_back(str.substr(start, end - start));
203
204         start = end + 1;
205     }
206
207     tokens.push_back(str.substr(start));
208
209     return tokens;
210 }
211
```