

You can view this report online at: https://www.hackerrank.com/x/tests/1121391/candidates/27774347/report

Full My Nguyen

Name:

Email: nguyen_my@yahoo.com

Test Strings and Arrays Assessment 2021

Name:

19 Jul 2021 17:05:36 PDT Taken

On:

88 min 27 sec/ 90 min Time

Taken:

+14084096862 Contact

Number:

Resume: https://hackerrank-

resumes.s3.amazonaws.com/412894/JhbK9vK_4Bhc4Gvuv7s5hgcFJGeFCAThWliNY1UGAfhwRPsrmVekT5ZtKXgX8QA2Ag/My_Nguyen_Resume.PDF

scored in Strings and Arrays

Assessment 2021 in 88 min 27 sec on 19 Jul 2021 17:05:36

49.1%

285/580

PDT

Linkedin: https://www.linkedin.com/in/my-nguyen-87849

Invited

Curriculum

by:

Skills Score:

Tags

Score:

Recruiter/Team Comments:

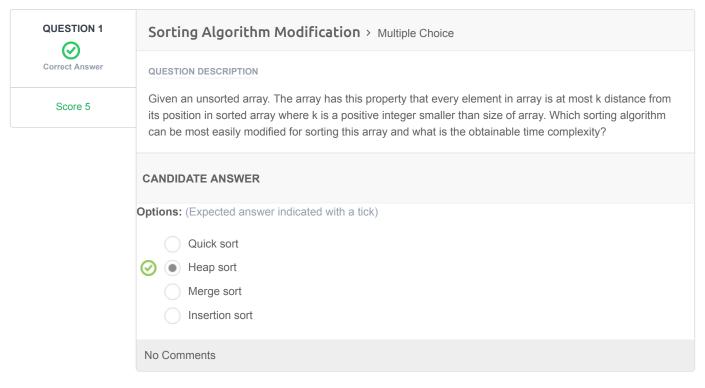
No Comments.

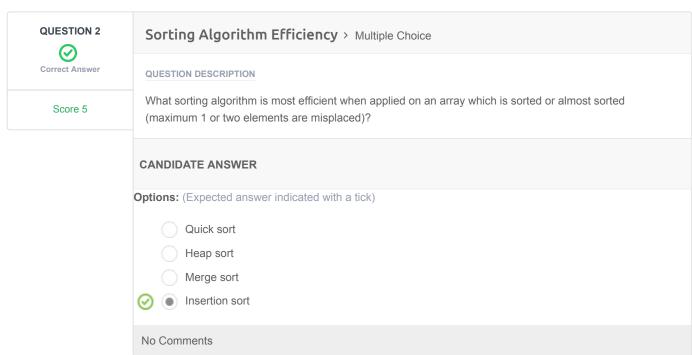
Plagiarism flagged

We have marked questions with suspected plagiarism below. Please review.

	Question Description	Time Taken	Score	Status
Q1	Sorting Algorithm Modification > Multiple Choice	1 min 9 sec	5/ 5	②
Q2	Sorting Algorithm Efficiency > Multiple Choice	46 sec	5/ 5	⊘
Q3	Runtime Analysis > Multiple Choice	2 min 2 sec	0/5	\otimes
Q4	Add Intervals Output > Multiple Choice	1 min 16 sec	5/ 5	\odot
Q5	Add Intervals Debugging > Coding	15 min 56 sec	100/ 100	(!)
Q6	Add Intervals Space Complexity > Multiple Choice	33 sec	5/ 5	⊘

Q7	Add Intervals Time Complexity > Multiple Choice	16 sec	5/ 5	⊘	
Q8	3Sum Closest > Coding	7 min 40 sec	140/ 150	⊘	
Q9	Rotate Image > Coding	25 min 37 sec	10/ 150	⊘	
Q10	Minimum Window Substring > Coding	33 min 1 sec	10/ 150	⊘	





QUESTION 3



Score 0

Runtime Analysis > Multiple Choice

QUESTION DESCRIPTION

What is the worst possible run time of this code? N refers to the size of the array "nums" and you can assume nums will be a sorted array.

Java:

```
int num occurences(ArrayList<Integer> nums, int x, int start, int end) {
    if (start > end) {
       return 0;
    int mid = (start + end) / 2;
    if (nums.get(mid) < x) {
        return num_occurences(nums, x, mid + 1, end);
    if (nums.get(mid) > x) {
        return num occurences (nums, x, start, mid - 1);
    return num occurences (nums, x, start, mid - 1) + 1 +
num occurences(nums, x, mid + 1, end);
```

Python:

```
def num_occurrences(nums, x, start, end):
  if start > end:
      return 0
  mid = (start + end) // 2
  if nums[mid] < x:</pre>
      return num occurrences (nums, x, mid + 1, end)
   if nums[mid] > x:
      return num occurrences (nums, x, start, mid - 1)
  return num occurrences (nums, x, start, mid - 1) + 1 +
num occurrences (nums, x, mid + 1, end)
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

$\Omega(1)$
O(1)
\ /

O(loc	'n



(n)



O(n log n)

$O(n^2)$
O(n^2)

No Comments

QUESTION 4



Score 5

Add Intervals Output > Multiple Choice

QUESTION DESCRIPTION

Add a new interval into a set of non-overlapping intervals, merging if necessary.

The intervals given will be sorted according to their start times.

Example-

Input: intervals = [[1,5],[6,12], [14, 15]], new_interval = [3,6]

Output: [[1, 12], [14, 15]]

Given the problem statement, what is the expected output for this input? add_intervals([[1,2],[3,4],[6,7],[8,10],[11,17]], [4,8])

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- [[1,2], [3, 8], [8,10], [11,17]]
- [[1,2], [3,4], [4, 8], [8,10], [11,17]]
- [[1,2], [3, 4], [4,10], [11,17]]



[11,17] (3, 10)

No Comments

QUESTION 5



Score 100

Add Intervals Debugging > Coding

QUESTION DESCRIPTION

Add a new interval into a set of non-overlapping intervals, merging if necessary.

The intervals given will be sorted according to their start times.

Example-

Input: intervals = [[1,5], [6,12], [14, 15]], new_interval = [3,6] Output: [[1, 12], [14, 15]]

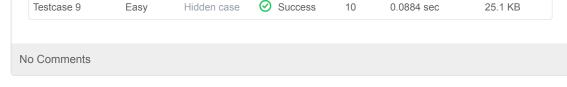
Please fix the buggy solution below.

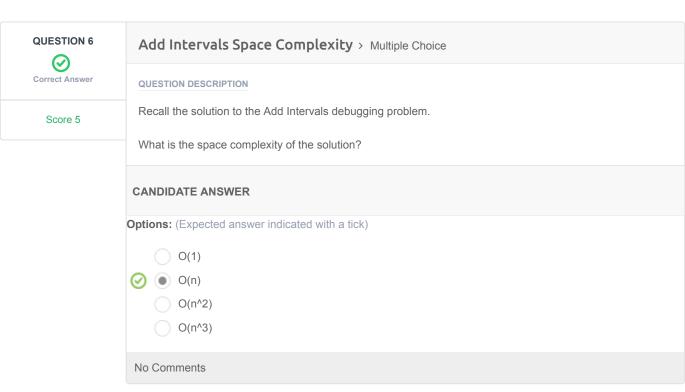
CANDIDATE ANSWER

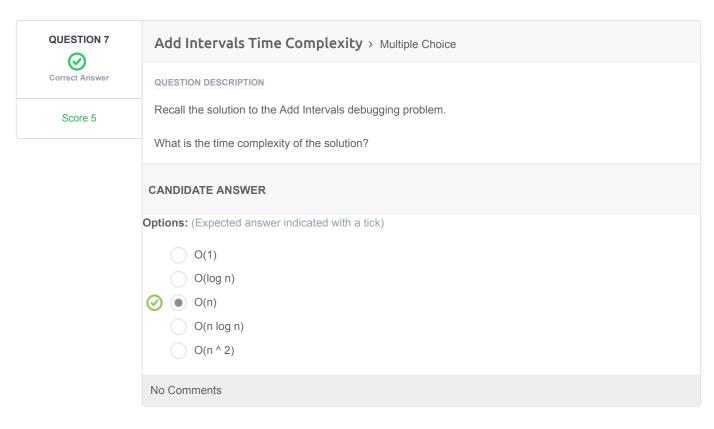
```
/**
      * Definition for an interval.
     * public class Interval {
     * int start;
           int end;
           Interval(int s, int e) { start = s; end = e; }
      * }
8
      */
     public static ArrayList<Interval> addInterval(ArrayList<Interval>
```

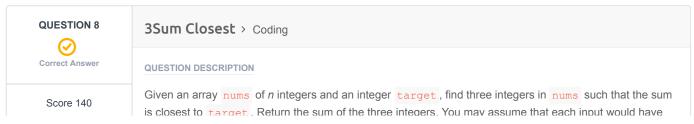
```
10 intervals, Interval newInterval) {
           ArrayList<Interval> result = new ArrayList<Interval>();
          int i = 0;
          while (i < intervals.size() && intervals.get(i).end <</pre>
14 newInterval.start) {
              result.add(intervals.get(i));
              i++;
          while (i < intervals.size() && intervals.get(i).start <=</pre>
19 newInterval.end) {
              newInterval.start = Math.min(newInterval.start,
21 intervals.get(i).start);
              newInterval.end = Math.max(newInterval.end,
23 intervals.get(i).end);
24
              i++;
         }
          result.add(newInterval);
          while (i < intervals.size()) {</pre>
             result.add(intervals.get(i));
              i++;
         return result;
          /*for (Interval interval : intervals) {
               if (interval.end < newInterval.start) {</pre>
                  result.add(interval);
                  System.out.println("SAVED interval: " + interval.start + ", "
37 + interval.end);
              } else {
                  newInterval = new Interval(interval.start,
40 Math.max(newInterval.end, interval.end));
                   System.out.println("NEW interval: " + newInterval.start + ",
42 " + newInterval.end);
              } else if (interval.start > newInterval.end) {
                  result.add(newInterval);
              } else if (interval.end >= newInterval.start || interval.start <=
46 newInterval.end) {
                  newInterval = new Interval(Math.min(interval.start,
   newInterval.start), Math.max(newInterval.end, interval.end));
           result.add(newInterval);
           System.out.println("LAST interval: " + newInterval.start + ", " +
   newInterval.end);*/
     }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.1192 sec	25 KB
Testcase 1	Easy	Hidden case	Success	10	0.1189 sec	25 KB
Testcase 2	Easy	Hidden case	Success	10	0.0884 sec	25.1 KB
Testcase 3	Easy	Hidden case	Success	10	0.0939 sec	25 KB
Testcase 4	Easy	Hidden case	Success	10	0.1152 sec	25.1 KB
Testcase 5	Easy	Hidden case	Success	10	0.103 sec	25.1 KB
Testcase 6	Easy	Hidden case	Success	10	0.0936 sec	25 KB
Testcase 7	Easy	Hidden case	Success	10	0.1034 sec	25.2 KB
Testcase 8	Easy	Hidden case	Success	10	0.1235 sec	25.3 KB









exactly one solution.

Example:

```
Given array nums = [-1, 2, 1, -4], and target = 1.

The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).
```

CANDIDATE ANSWER

```
public static int threeSumClosest(int[] nums, int target) {
         Arrays.sort(nums);
          int min = Integer.MAX_VALUE;
4
          for (int i = 0; i < nums.length-2; i++) {
              int left = i + 1;
              int right = nums.length-1;
8
              while (left < right) {
                  int sum = nums[i] + nums[left] + nums[right];
                  if (sum < target)</pre>
                       left++;
                  else if (sum > target)
                      right--;
14
                   else
                      return sum;
                  if (Math.abs(sum-target) < Math.abs(min-target))</pre>
                      min = sum;
              }
         }
          return min;
24
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.1293 sec	24.9 KB
Testcase 1	Easy	Hidden case	Success	10	0.1139 sec	25.1 KB
Testcase 2	Easy	Hidden case	Success	10	0.0948 sec	24.9 KB
Testcase 3	Easy	Hidden case	Success	10	0.1366 sec	25 KB
Testcase 4	Easy	Hidden case	Success	10	0.0939 sec	24.7 KB
Testcase 5	Easy	Hidden case	Success	10	0.0906 sec	25.1 KB
Testcase 6	Easy	Hidden case	Success	10	0.0953 sec	24.9 KB
Testcase 7	Easy	Hidden case	Success	10	0.1257 sec	25 KB
Testcase 8	Easy	Hidden case	Success	10	0.1017 sec	24.8 KB
Testcase 9	Easy	Hidden case	Success	10	0.0884 sec	24.9 KB
Testcase 10	Easy	Hidden case	Success	10	0.0959 sec	24.9 KB
Testcase 11	Easy	Hidden case	Success	10	0.1062 sec	24.9 KB
Testcase 12	Easy	Hidden case	Success	10	0.0904 sec	25.1 KB
Testcase 13	Easy	Hidden case	Wrong Answer	0	0.1346 sec	24.9 KB

No Comments

QUESTION 9



Score 10

Rotate Image > Coding

QUESTION DESCRIPTION

You are given an $n \times n$ 2D matrix representing an image. Rotate the image by 90 degrees (clockwise).

Note:

You have to rotate the image **in-place**, which means you have to modify the input 2D matrix directly. **DO NOT**allocate another 2D matrix and do the rotation.

Example 1:

```
Given input matrix =
[
    [1,2,3],
    [4,5,6],
    [7,8,9]
],

rotate the input matrix in-place such that it becomes:
[
    [7,4,1],
    [8,5,2],
    [9,6,3]
]
```

Example 2:

```
Given input matrix =

[
    [5, 1, 9,11],
    [2, 4, 8,10],
    [13, 3, 6, 7],
    [15,14,12,16]
],

rotate the input matrix in-place such that it becomes:

[
    [15,13, 2, 5],
    [14, 3, 4, 1],
    [12, 6, 8, 9],
    [16, 7,10,11]
]
```

CANDIDATE ANSWER

```
public static void rotate(int[][] matrix) {
   int N = matrix.length;
   int top = 0;
   int right = N-1;
   int bottom = N-1;
```

```
int left = 0;
           /*while (true)*/ {
8
              int[] tmp1 = new int[N];
9
               for (int i = 0; i < N; i++) {
                   tmp1[i] = matrix[i][right];
                   matrix[i][right] = matrix[top][i];
               for (int i = bottom; i >= 0; i--) {
14
                   matrix[top][bottom-i] = matrix[i][left];
               }
               for (int i = right; i \ge 0; i--)
                   matrix[i][left] = matrix[bottom][i];
               for (int i = 0; i < bottom; i++) {
19
                   matrix[bottom][bottom-i] = tmp1[i];
           }
       }
24
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Wrong Answer	0	0.1279 sec	25 KB
Testcase 1	Easy	Sample case	Wrong Answer	0	0.097 sec	25.1 KB
Testcase 2	Easy	Hidden case	Wrong Answer	0	0.0952 sec	25.1 KB
Testcase 3	Easy	Hidden case	Wrong Answer	0	0.0985 sec	24.9 KB
Testcase 4	Easy	Hidden case	Wrong Answer	0	0.0937 sec	24.9 KB
Testcase 5	Easy	Hidden case	Wrong Answer	0	0.0965 sec	24.9 KB
Testcase 6	Easy	Hidden case	Wrong Answer	0	0.0977 sec	25 KB
Testcase 7	Easy	Hidden case	Wrong Answer	0	0.1245 sec	24.8 KB
Testcase 8	Easy	Hidden case	Wrong Answer	0	0.103 sec	25 KB
Testcase 9	Easy	Hidden case	Success	10	0.0892 sec	24.9 KB
Testcase 10	Easy	Hidden case	Wrong Answer	0	0.127 sec	25 KB
Testcase 11	Easy	Hidden case	Wrong Answer	0	0.0956 sec	24.9 KB
Testcase 12	Easy	Hidden case	Wrong Answer	0	0.0967 sec	25 KB
Testcase 13	Easy	Hidden case	Wrong Answer	0	0.0865 sec	25 KB
Testcase 14	Easy	Hidden case	⊗ Wrong Answer	0	0.1273 sec	25 KB

No Comments

QUESTION 10



Correct Answer

Score 10

Minimum Window Substring > Coding

QUESTION DESCRIPTION

Given a string S and a string T, find the minimum window in S which will contain all the characters in T in complexity O(n).

Example:

```
Input: S = "ADOBECODEBANC", T = "ABC"
```

Output: "BANC"

Note:

- If there is no such window in S that covers all characters in T, return the empty string "".
- If there is such window, you are guaranteed that there will always be only one unique minimum window in S

CANDIDATE ANSWER

```
public static String minWindow(String s, String t) {
          Map<Character, Integer> map = new HashMap<>();
          for (char c : t.toCharArray()) {
              int count = map.getOrDefault(c, 0);
               map.put(c, count+1);
8
          int start = -1;
          int end = -1;
          String result = s;
          int min = Integer.MAX_VALUE;
          for (int i = 0; i < s.length(); i++) {
              char c = s.charAt(i);
14
               if (map.containsKey(c)) {
                  // ?????
                  int count = map.get(c);
                   if (count != 0) {
                       map.put(c, count-1);
                       if (start == -1)
                           start = i;
                   } else {
                       boolean isMapEmpty = false;
                       int tmp = 0;
                      for (int value : map.values()) {
                           if (value == 0)
                               tmp++;
                       if (tmp == map.size()) {
                           end = i;
                           if (end-start+1 < min) {</pre>
                              min = end-start+1;
                               result = s.substring(start, end+1);
                           }
34
                       }
                   }
          return result;
       }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TESTUASE	DIFFICULIT	TTPE	STATUS	SCURE	TIVIE TAKEN	WEWORT USED
Testcase 0	Easy	Sample case	Wrong Answer	0	0.0959 sec	24.7 KB
Testcase 2	Easy	Hidden case	Success	10	0.1266 sec	24.9 KB
Testcase 3	Easy	Hidden case	Wrong Answer	0	0.0904 sec	24.8 KB
Tactosca /	Facu	Hidden case	Mrong Anewer	Λ	0 0872 000	24 0 KB

Testease 4	Lasy	Hidden Case	Wrong Answei	0	0.0072 300	24.3 ND
Testcase 5	Easy	Hidden case	⊗ Wrong Answer	0	0.1179 sec	24.9 KB
Testcase 6	Easy	Hidden case	⊗ Wrong Answer	0	0.0874 sec	25.1 KB
Testcase 7	Easy	Hidden case	⊗ Wrong Answer	0	0.0902 sec	25 KB
Testcase 8	Easy	Hidden case	⊗ Wrong Answer	0	0.103 sec	25 KB
Testcase 9	Easy	Hidden case	⊗ Wrong Answer	0	0.1268 sec	25.1 KB
Testcase 10	Easy	Hidden case	Wrong Answer	0	0.0977 sec	25.2 KB
Testcase 11	Easy	Hidden case	Wrong Answer	0	0.1044 sec	24.8 KB
Testcase 12	Easy	Hidden case	Wrong Answer	0	0.1066 sec	24.9 KB
Testcase 13	Easy	Hidden case	Wrong Answer	0	0.1287 sec	24.9 KB
Testcase 14	Easy	Hidden case	Wrong Answer	0	0.0966 sec	25 KB
Testcase 15	Easy	Hidden case	Wrong Answer	0	0.1021 sec	25 KB

PDF generated at: 20 Jul 2021 01:36:05 UTC