



You can view this report online at : <https://www.hackerrank.com/x/tests/1118667/candidates/26560399/report>

Full Name:

My Nguyen

Email:

nguyen_my@yahoo.com

Test Name:

Core Data Structures Assessment 2021

Taken On:

10 Jun 2021 16:19:45 PDT

Time Taken:

88 min 39 sec/ 90 min

Contact Number:

+14084096862

Resume:

https://hackerrank-resumes.s3.amazonaws.com/412894/JhbK9vK_4Bhc4Gvuv7s5hgcFJGeFCATHWliINY1UGAfhwRPsrnVekT5ZtKXgX8QA2Ag/My_Nguyen_Resume.PDF

Linkedin:

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

Invited by:

Curriculum

Skills Score:

Tags Score:

40.8%

200/490

scored in **Core Data Structures Assessment 2021** in 88 min 39 sec on 10 Jun 2021 16:19:45 PDT

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Min Heap Array > Multiple Choice	7 min 31 sec	5/ 5	✓
Q2	Max Heap Array > Multiple Choice	2 min 7 sec	5/ 5	✓
Q3	Well-Formed Brackets, Part 1 > Multiple Choice	1 min 7 sec	5/ 5	✓
Q4	Well-Formed Brackets, Part 2 > Coding	10 min 42 sec	60/ 60	✓
Q5	Well-Formed Brackets, Part 3 > Multiple Choice	38 sec	5/ 5	✓
Q6	Well-Formed Brackets, Part 4 > Multiple Choice	15 sec	5/ 5	✓
Q7	Almost Sorted Array, Part 1 > Coding	5 min 39 sec	100/ 100	✓
Q8	Almost Sorted Array, Part 2 > Multiple Choice	37 sec	5/ 5	✓
Q9	Simplify Path > Coding	37 min 20 sec	10/ 150	⚠



QUESTION 1



Correct Answer

Score 5

Min Heap Array > Multiple Choice

QUESTION DESCRIPTION

What would be the array representation of this min heap after these operations? Utilize the operations defined in the guide.

Java:

```
Heap h = new Heap(10); // Creates a min-heap with 10 as the root
h.insert(8);
h.insert(5);
h.insert(1);
h.insert(6);
h.insert(2);
h.removeMin();
```

Python:

```
starterList = [10]
h = heapq.heapify(starterList) # Creates a min-heap with 10 as the root
h.insert(8)
h.insert(5)
h.insert(1)
h.insert(6)
h.insert(2)
h.removeMin()
```

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ [2, 6, 5, 10, 8]
- ☒ [2, 5, 8, 10, 6]
- ☐ [2, 6, 5, 8, 10]
- ☐ [2, 5, 6, 10, 8]

No Comments

QUESTION 2

Correct Answer

Score 5

Max Heap Array > Multiple Choice**QUESTION DESCRIPTION**

What would be the array representation of this max heap after these operations? Utilize the operations defined in the guide.

Java:

```
Heap h = new Heap(10); // Creates a max-heap with 10 as the root
h.insert(8);
h.insert(5);
h.insert(1);
h.insert(6);
h.insert(2);
h.removeMax();
```

Python:

```
starterList = [10]
h = heapq.heapify(starterList) # Creates a max-heap with 10 as the root
h.insert(8)
h.insert(5)
h.insert(1)
h.insert(6)
h.insert(2)
h.removeMax()
```

CANDIDATE ANSWER**Options:** (Expected answer indicated with a tick)

- ☐ [8, 6, 5, 2, 1]
- ☐ [8, 5, 6, 2, 1]
- ☒ [8, 6, 5, 1, 2]
- ☐ [8, 5, 6, 1, 2]

No Comments

QUESTION 3



Correct Answer

Score 5

Well-Formed Brackets, Part 1 > Multiple Choice

QUESTION DESCRIPTION

A string consisting of only these characters: “(”, “)”, “[”, “]”, “{”, and “}” is considered well formed if the different types of brackets are matched in the correct order. For example, “{)” would not be well formed and “[(){}]” would be considered well formed. Write a program that returns true if a given string is well-formed and false otherwise.

What would be the expected output of this input, “[(){}()]”?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ true
- ☒ false

No Comments

QUESTION 4



Correct Answer

Score 60

Well-Formed Brackets, Part 2 > Coding

QUESTION DESCRIPTION

A string consisting of only these characters: “(”, “)”, “[”, “]”, “{”, and “}” is considered well formed if the different types of brackets are matched in the correct order. For example, “{)” would not be well formed and “[(){}]” would be considered well formed. Write a program that true if a given string is well-formed and false otherwise.

The snippet below is a buggy solution to the problem above. Fix the buggy solution such that it solves the problem.

CANDIDATE ANSWER

Language used: Java 7

```
1      public static boolean isValid(String s) {
2          HashMap<Character, Character> map = new HashMap<Character, Character>
3      ();
4          map.put('(', ')');
5          map.put('[', ']');
6          map.put('{', '}');
7
8          Stack<Character> stack = new Stack<Character>();
9
10         for (int i = 0; i < s.length(); i++) {
11             char curr = s.charAt(i);
12
13             if (map.containsKey(curr)) {
14                 stack.push(curr);
15             } else if (stack.isEmpty() || map.get(stack.pop()) != curr) {
16                 return false;
17             }
18         }
19
20         return stack.empty();
```

```
20         return student.empty();  
21     }  
22 }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✔ Success	0	0.0729 sec	23.4 KB
Testcase 1	Easy	Sample case	✔ Success	10	0.0642 sec	23.7 KB
Testcase 2	Easy	Hidden case	✔ Success	0	0.0728 sec	23.5 KB
Testcase 3	Easy	Hidden case	✔ Success	0	0.0842 sec	23.6 KB
Testcase 4	Easy	Hidden case	✔ Success	0	0.0653 sec	23.7 KB
Testcase 5	Easy	Hidden case	✔ Success	10	0.0728 sec	23.5 KB
Testcase 6	Easy	Hidden case	✔ Success	0	0.0889 sec	23.6 KB
Testcase 7	Easy	Hidden case	✔ Success	10	0.0638 sec	23.5 KB
Testcase 8	Easy	Hidden case	✔ Success	0	0.0691 sec	23.5 KB
Testcase 9	Easy	Hidden case	✔ Success	10	0.0684 sec	23.7 KB
Testcase 10	Easy	Hidden case	✔ Success	10	0.0646 sec	23.7 KB
Testcase 11	Easy	Hidden case	✔ Success	10	0.0667 sec	23.7 KB
Testcase 12	Easy	Hidden case	✔ Success	0	0.0715 sec	23.7 KB
Testcase 13	Easy	Hidden case	✔ Success	0	0.0663 sec	23.6 KB
Testcase 14	Easy	Hidden case	✔ Success	0	0.0798 sec	23.7 KB

No Comments

QUESTION 5



Correct Answer

Score 5

Well-Formed Brackets, Part 3 > Multiple Choice

QUESTION DESCRIPTION

A string consisting of only these characters: “(”, “)”, “[”, “]”, “{”, and “}” is considered well formed if the different types of brackets are matched in the correct order. For example, “{)” would not be well formed and “[()]” would be considered well formed. Write a program that true if a given string is well-formed and false otherwise.

What is the runtime of the algorithm once it is fixed? n represents the size of the input string.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ $O(1)$
- ☐ $O(\log n)$
- ☒ $O(n)$
- ☐ $O(n \log n)$
- ☐ $O(n^2)$

No Comments

QUESTION 6



Correct Answer

Score 5

Well-Formed Brackets, Part 4 > Multiple Choice

QUESTION DESCRIPTION

A string consisting of only these characters: “(”, “)”, “[”, “]”, “{”, and “}” is considered well formed if the different types of brackets are matched in the correct order. For example, “{)” would not be well formed and “[(){}]” would be considered well formed. Write a program that true if a given string is well-formed and false otherwise.

What is the space complexity of the algorithm once it is fixed? n represents the size of the input string.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ $O(1)$
- ☐ $O(\log n)$
- ☒ $O(n)$
- ☐ $O(n \log n)$
- ☐ $O(n^2)$

No Comments

QUESTION 7



Correct Answer

Score 100

Almost Sorted Array, Part 1 > Coding

QUESTION DESCRIPTION

Given an almost sorted array, in which each number is less than m spots away from its correctly sorted position, and the value m , write an algorithm that will return an array with the elements properly sorted.

An example input would be the list [3, 2, 1, 4, 6, 5] and $m = 3$. In this example, each element in the array is less than 3 spots away from its position in a sorted array.

The snippet below is a buggy solution to the problem above. Fix the buggy solution such that it solves the problem.

CANDIDATE ANSWER

Language used: Java 7

```

1      public static ArrayList<Integer> sortList(Integer[] almostSortedList, int
2  m) {
3          PriorityQueue<Integer> minHeap = new PriorityQueue<Integer>();
4          ArrayList<Integer> result = new ArrayList<Integer>();
5
6          /*for (int i = 0; i < m; i++) {
7              minHeap.offer(almostSortedList[i]);
8          }*/
9
10         for (int i = 0; i < almostSortedList.length; i++) {
11             minHeap.offer(almostSortedList[i]);
12             if (minHeap.size() > m)
13                 result.add(minHeap.poll());

```

```

14     }
15
16     while (minHeap.peek() != null) {
17         result.add(minHeap.poll());
18     }
19
20     return result;
21 }
22

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	✔ Success	10	0.0659 sec	23.7 KB
Testcase 1	Easy	Hidden case	✔ Success	10	0.0633 sec	23.7 KB
Testcase 2	Easy	Hidden case	✔ Success	10	0.0732 sec	23.9 KB
Testcase 3	Easy	Hidden case	✔ Success	10	0.082 sec	23.9 KB
Testcase 4	Easy	Hidden case	✔ Success	10	0.0666 sec	23.7 KB
Testcase 5	Easy	Hidden case	✔ Success	10	0.0748 sec	23.9 KB
Testcase 6	Easy	Hidden case	✔ Success	10	0.0723 sec	23.8 KB
Testcase 7	Easy	Hidden case	✔ Success	10	0.074 sec	23.8 KB
Testcase 8	Easy	Hidden case	✔ Success	10	0.0713 sec	23.9 KB
Testcase 9	Easy	Hidden case	✔ Success	10	0.0642 sec	23.7 KB

No Comments

QUESTION 8



Correct Answer

Score 5

Almost Sorted Array, Part 2 > Multiple Choice

QUESTION DESCRIPTION

Given an almost sorted array, in which each number is less than m spots away from its correctly sorted position, and the value m , write an algorithm that will return an array with the elements properly sorted.

An example input would be the list $[3, 2, 1, 4, 6, 5]$ and $m = 3$. In this example, each element in the array is less than 3 spots away from its position in a sorted array.

What is the runtime of the algorithm once it is fixed? n is the length of the array.

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ☐ $O(\log n)$
- ☐ $O(\log m)$
- ☐ $O(n)$
- ✔ ☒ $O(n \log m)$
- ☐ $O(m \log n)$

No Comments



Simplify Path > Coding

QUESTION DESCRIPTION

Given an absolute path for a file (Unix-style), simplify it.

For example,

path = "/home/", => "/home"

path = "/a/./b/../../c/", => "/c"

Corner Cases:

- Did you consider the case where path = "/. ./"? In this case, you should return "/".
- Another corner case is the path might contain multiple slashes "/" together, such as "/home//foo/". In this case, you should ignore redundant slashes and return "/home/foo".

CANDIDATE ANSWER

Language used: Java 7

```

1      public static String simplifyPath(String path) {
2          if (path.equals("/./"))
3              return "/";
4
5          Stack<String> stack = new Stack<>();
6          int i = 0;
7          boolean atEnd = false;
8          while (i < path.length() && !atEnd) {
9              switch (path.charAt(i)) {
10                 case '.':
11                     if (path.charAt(i) == '.') {
12                         if (path.charAt(i+1) == '.' && path.charAt(i+1) ==
13 '/' ) {
14                             System.out.println("case ../, i: " + i);
15                             stack.pop();
16                             i += 3;
17                         } else if (path.charAt(i+1) == '/') {
18                             System.out.println("case ./, i: " + i);
19                             i += 2;
20                         }
21                     }
22                     break;
23                 case '/':
24                     if (i == path.length()-1) {
25                         atEnd = true;
26                         System.out.println("at end i: " + i);
27                     } else if (path.charAt(i+1) == '/') {
28                         System.out.println("case //, i: " + i);
29                         i += 2;
30                     } /*} else if (path.charAt(i+1) == '.') {
31                         */
32                     } else {
33                         System.out.println("case /, i: " + i);
34                         i += 1;
35                     }
36                     break;
37                 default:
38                     StringBuilder folder = new StringBuilder();
39                     while (i < path.length() && path.charAt(i) != '/') {
40                         folder.append(path.charAt(i));
41                         i++;
42                     }

```



```

43         System.out.println("folder: " + folder.toString() + ", i:
44         " + i);
45         stack.push(folder.toString());
46     }
47 }
48 StringBuilder sb = new StringBuilder("");
49 while (!stack.isEmpty()) {
50     sb.insert(0, stack.pop());
51     sb.insert(0, "/");
52 }
53 return sb.toString();
54 }

```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	❌ Wrong Answer	0	0.0653 sec	23.7 KB
Testcase 1	Easy	Sample case	❌ Terminated due to timeout	0	4.0034 sec	26 KB
Testcase 2	Easy	Sample case	✅ Success	10	0.0645 sec	23.6 KB
Testcase 3	Easy	Sample case	❌ Wrong Answer	0	0.0709 sec	23.5 KB
Testcase 4	Easy	Hidden case	❌ Terminated due to timeout	0	4.0036 sec	26 KB
Testcase 5	Easy	Hidden case	❌ Terminated due to timeout	0	4.0036 sec	25.9 KB
Testcase 6	Easy	Hidden case	❌ Terminated due to timeout	0	4.003 sec	26.1 KB
Testcase 7	Easy	Hidden case	❌ Terminated due to timeout	0	4.0033 sec	25.8 KB
Testcase 8	Easy	Hidden case	❌ Wrong Answer	0	0.075 sec	23.8 KB
Testcase 9	Easy	Hidden case	❌ Terminated due to timeout	0	4.0038 sec	26.1 KB
Testcase 10	Easy	Hidden case	❌ Terminated due to timeout	0	4.003 sec	26 KB
Testcase 11	Easy	Hidden case	❌ Terminated due to timeout	0	4.0033 sec	26 KB
Testcase 12	Easy	Hidden case	❌ Wrong Answer	0	0.0655 sec	23.6 KB
Testcase 13	Easy	Hidden case	❌ Runtime Error	0	0.0702 sec	23.6 KB
Testcase 14	Easy	Hidden case	❌ Wrong Answer	0	0.0695 sec	23.7 KB

No Comments

QUESTION 10



Wrong Answer

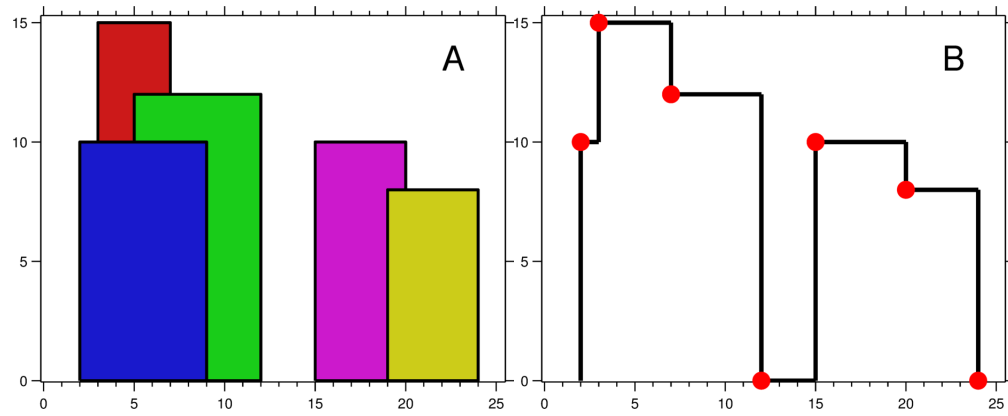
Score 0

Skyline Problem > Coding

QUESTION DESCRIPTION

A city's skyline is the outer contour of the silhouette formed by all the buildings in that city when viewed from

a distance. Now suppose you are given the locations and height of all the buildings as shown on a cityscape photo (Figure A), write a program to **output the skyline** formed by these buildings collectively (Figure B).



The geometric information of each building is represented by a triplet of integers $[Li, Ri, Hi]$, where Li and Ri are the x coordinates of the left and right edge of the i th building, respectively, and Hi is its height. It is guaranteed that $0 \leq Li, Ri \leq INT_MAX$, $0 < Hi \leq INT_MAX$, and $Ri - Li > 0$. You may assume all buildings are perfect rectangles grounded on an absolutely flat surface at height 0.

For instance, the dimensions of all buildings in Figure A are recorded as: $[[2, 9, 10], [3, 7, 15], [5, 12, 12], [15, 20, 10], [19, 24, 8]]$.

The output is a list of "key points" (red dots in Figure B) in the format of $[[x1, y1], [x2, y2], [x3, y3], \dots]$ that uniquely defines a skyline. **A key point is the left endpoint of a horizontal line segment.** Note that the last key point, where the rightmost building ends, is merely used to mark the termination of the skyline, and always has zero height. Also, the ground in between any two adjacent buildings should be considered part of the skyline contour.

For instance, the skyline in Figure B should be represented as: $[[2, 10], [3, 15], [7, 12], [12, 0], [15, 10], [20, 8], [24, 0]]$.

Notes:

- The number of buildings in any input list is guaranteed to be in the range $[0, 10000]$.
- The input list is already sorted in ascending order by the left x position Li .
- The output list must be sorted by the x position.
- There must be no consecutive horizontal lines of equal height in the output skyline. For instance, $[[\dots[2, 3], [4, 5], [7, 5], [11, 5], [12, 7], \dots]]$ is not acceptable; the three lines of height 5 should be merged into one in the final output as such: $[[\dots[2, 3], [4, 5], [12, 7], \dots]]$

CANDIDATE ANSWER

Language used: **Java 7**

```
1 public static List<int[]> getSkyline(int[][] buildings) {
2     /*Map<Int, Int> map = new TreeMap<>();
3     for (int[] building : buildings) {
4         int left = building[0];
5         int right = building[1];
6         int height = building[2];
7         int cLeft = map.getDefault(left, 0);
8         map[left] = Math.max(cLeft, height);
9         int cRight = map.getDefault(right, 0);
10        map[right] = Math.max(cRight, height);
11    }*/
12    Queue<int[]> maxHeap = new PriorityQueue<>(new Comparator<int[]>() {
13        @Override
14        public int compare(int[] a, int[] b) {
15            return b[2] - a[2];
16        }
17    });
18    for (int[] building : buildings) {
19        int left = building[0];
20        int right = building[1];
21        int height = building[2];
22        while (!maxHeap.isEmpty() && maxHeap.peek()[1] < left) {
23            maxHeap.poll();
24        }
25        while (!maxHeap.isEmpty() && maxHeap.peek()[2] < height) {
26            maxHeap.poll();
27        }
28        maxHeap.offer(building);
29    }
30    List<int[]> result = new ArrayList<>();
31    int prevHeight = 0;
32    while (!maxHeap.isEmpty()) {
33        int[] cur = maxHeap.poll();
34        int curLeft = cur[0];
35        int curRight = cur[1];
36        int curHeight = cur[2];
37        if (prevHeight < curHeight) {
38            result.add(new int[] {curLeft, curHeight});
39            prevHeight = curHeight;
40        }
41        if (curRight < result.getLast()[0]) {
42            result.add(new int[] {curRight, 0});
43        }
44    }
45    return result;
46 }
```

```
15         return b[2] - b[1];
16     }
17     });
18     for (int[] building : buildings) {
19         maxHeap.add(building);
20     }
21
22     List<int[]> result = new ArrayList<>();
23     while (!maxHeap.isEmpty()) {
24         int[] top = maxHeap.peak();
25     }
26     return null;
27 }
28
```

Result: Compilation Failed

Compile Message

```
Solution.java:19: error: cannot infer type arguments for
PriorityQueue<>
    Queue maxHeap = new PriorityQueue<>(new Comparator() {
                                ^
1 error
```

No Comments