You can view this report online at : https://www.hackerrank.com/x/tests/1330875/candidates/41196773/report

Full Name:	Ruchit Bhardwaj
Email:	ruchitbh@usc.edu
Test Name:	**CodePath SE103: Unit 2 Assessment - Summer 2022**
Taken On:	14 Jun 2022 17:51:50 PDT
Time Taken:	84 min 34 sec/ 90 min
Invited by:	CodePath
Skills Score:	
Tags Score:	

69.4%

340/490

scored in **CodePath SE103: Unit 2 Assessment - Summer 2022** in 84 min 34 sec on 14 Jun 2022 17:51:50 PDT

Recruiter/Team Comments:

No Comments.

Question Description	Time Taken	Score	Status
Q1 Min Heap Array > Multiple Choice	6 min 47 sec	5/ 5	✓
Q2 Max Heap Array > Multiple Choice	4 min 4 sec	5/ 5	✓
Q3 Well-Formed Brackets, Part 1 > Multiple Choice	2 min 9 sec	5/ 5	✓
Q4 Well-Formed Brackets, Part 2 > Coding	16 min 47 sec	60/ 60	✓
Q5 Well-Formed Brackets, Part 3 > Multiple Choice	21 sec	5/ 5	✓
Q6 Well-Formed Brackets, Part 4 > Multiple Choice	26 sec	5/ 5	✓
Q7 Almost Sorted Array, Part 1 > Coding	9 min 47 sec	100/ 100	✓
Q8 Almost Sorted Array, Part 2 > Multiple Choice	1 min 10 sec	5/ 5	✓
Q9 Simplify Path > Coding	39 min 39 sec	150/ 150	✓
Q10 Challenge: Skyline Problem > Coding	3 min 10 sec	0/ 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          Stack<Character> stack = new Stack<>();
3          for (char c: s.toCharArray()) {
4              if (c == '(')
5                  stack.push(')');
6              else if (c == '[')
7                  stack.push(']');
8              else if (c == '{')
9                  stack.push('}');
10             else if (stack.isEmpty() || stack.pop() != c)
11                 return false;
12         }
13         return stack.isEmpty();
14     }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	0	0.0783 sec	23.5 KB
Testcase 1	Easy	Sample case	Success	10	0.0875 sec	23.4 KB
Testcase 2	Easy	Hidden case	Success	0	0.1025 sec	23.6 KB
Testcase 3	Easy	Hidden case	Success	0	0.0756 sec	23.7 KB
Testcase 4	Easy	Hidden case	Success	0	0.1071 sec	23.8 KB
Testcase 5	Easy	Hidden case	Success	10	0.1068 sec	23.5 KB
Testcase 6	Easy	Hidden case	Success	0	0.0837 sec	23.7 KB
Testcase 7	Easy	Hidden case	Success	10	0.119 sec	23.8 KB
Testcase 8	Easy	Hidden case	Success	0	0.0976 sec	23.6 KB
Testcase 9	Easy	Hidden case	Success	10	0.0705 sec	23.9 KB
Testcase 10	Easy	Hidden case	Success	10	0.0802 sec	23.7 KB
Testcase 11	Easy	Hidden case	Success	10	0.089 sec	23.8 KB
Testcase 12	Easy	Hidden case	Success	0	0.0885 sec	23.5 KB
Testcase 13	Easy	Hidden case	Success	0	0.0988 sec	23.7 KB
Testcase 14	Easy	Hidden case	Success	0	0.1107 sec	23.6 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 8**

```
    public static ArrayList<Integer> sortList(Integer[] almostSortedList, int
m) {
        PriorityQueue<Integer> minHeap = new PriorityQueue<Integer>();
        ArrayList<Integer> result = new ArrayList<Integer>();

        for (int i = 0; i < m; i++) {
            minHeap.offer(almostSortedList[i]);
        }

        for (int i = m; i < almostSortedList.length; i++) {
            minHeap.offer(almostSortedList[i]);
            result.add(minHeap.poll());
        }

        while (minHeap.peek() != null) {
            result.add(minHeap.poll());
        }

        return result;
    }
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.1304 sec	24.9 KB
Testcase 1	Easy	Hidden case	Success	10	0.1055 sec	24.8 KB
Testcase 2	Easy	Hidden case	Success	10	0.089 sec	25.1 KB
Testcase 3	Easy	Hidden case	Success	10	0.121 sec	25.1 KB
Testcase 4	Easy	Hidden case	Success	10	0.2328 sec	25 KB
Testcase 5	Easy	Hidden case	Success	10	0.1345 sec	25.1 KB
Testcase 6	Easy	Hidden case	Success	10	0.1204 sec	25 KB
Testcase 7	Easy	Hidden case	Success	10	0.1266 sec	25 KB
Testcase 8	Easy	Hidden case	Success	10	0.1266 sec	24.9 KB
Testcase 9	Easy	Hidden case	Success	10	0.1116 sec	24.8 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

QUESTION 9

Correct Answer

Score 150

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 8**

```
public static String simplifyPath(String path) {
    String[] arr = path.split("(\\/)+");
    Stack<String> stack = new Stack<>();
    for (int i = 0; i < arr.length; i++) {
        if (!stack.empty() && arr[i].equals("..")) {
            stack.pop();
        } else if (!arr[i].equals("") && !arr[i].equals(".") &&
!arr[i].equals("..")) {
            stack.push(arr[i]);
```

```
    }

    if (stack.empty()) {
        return "/";
    }

    StringBuilder sb = new StringBuilder();
    while (!stack.empty()) {
        sb.insert(0, stack.pop()).insert(0, "/");
    }
    return sb.toString();

}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 0	Easy	Sample case	Success	10	0.1458 sec	25 KB
Testcase 1	Easy	Sample case	Success	10	0.1153 sec	25 KB
Testcase 2	Easy	Sample case	Success	10	0.091 sec	25.1 KB
Testcase 3	Easy	Sample case	Success	10	0.1116 sec	24.9 KB
Testcase 4	Easy	Hidden case	Success	10	0.1304 sec	25 KB
Testcase 5	Easy	Hidden case	Success	10	0.0876 sec	24.9 KB
Testcase 6	Easy	Hidden case	Success	10	0.1016 sec	24.6 KB
Testcase 7	Easy	Hidden case	Success	10	0.1004 sec	24.9 KB
Testcase 8	Easy	Hidden case	Success	10	0.1165 sec	25 KB
Testcase 9	Easy	Hidden case	Success	10	0.083 sec	24.8 KB
Testcase 10	Easy	Hidden case	Success	10	0.1157 sec	24.8 KB
Testcase 11	Easy	Hidden case	Success	10	0.1378 sec	25 KB
Testcase 12	Easy	Hidden case	Success	10	0.1146 sec	24.9 KB
Testcase 13	Easy	Hidden case	Success	10	0.1307 sec	24.9 KB
Testcase 14	Easy	Hidden case	Success	10	0.1099 sec	24.8 KB

No Comments

QUESTION 10

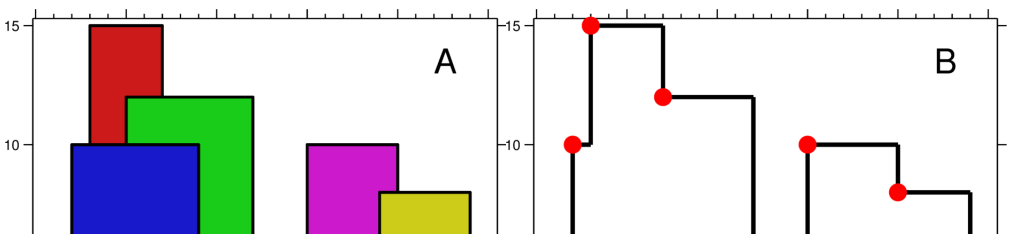
Wrong Answer

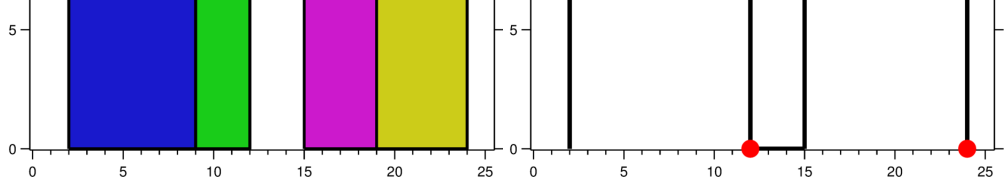
Score 0

Challenge: 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 ith building, respectively, and `Hi` is its height. It is guaranteed that `0 ≤ Li, Ri ≤ INT_MAX, 0 < Hi ≤ 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], ... ]` 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, `[...[2 3], [4 5], [7 5], [11 5], [12 7]...]` is not acceptable; the three lines of height 5 should be merged into one in the final output as such: `[...[2 3], [4 5], [12 7], ...]`

CANDIDATE ANSWER

Language used: **Java 8**

```
1  public static List<int[]> getSkyline(int[][] buildings) {
2
3  }
4
5
```

Result: Compilation Failed

Compile Message

```
Solution.java:11: error: missing return statement
}
^
1 error
```

No Comments