6 kyu Largest Radial Sum

🖒 10 📚 1 🛷 88% of 52 🔘 87 of 235 💄 zeroxoneafour

Instructions

Output

Time: 755ms Passed: 3 Failed: 0

Test Results:

Fixed Tests

regular

edge cases

Completed in 1ms

You have passed all of the tests!:)

Random Tests

> random

Completed in 4ms

JavaScript

Node v18.x

♥

Solution

```
function largestRadialSum(arr, d) {
    let honorSum;
    let honorBiggestSum = Number.NEGATIVE_INFINITY;
    for (let i = 0; i < arr.length/d; i++) {
        honorSum = 0;
        for (let j = 0; j < arr.length; j += arr.length/d) {
            honorSum += arr[i + j];
        }
        if(honorBiggestSum < honorSum){
            honorBiggestSum = honorSum;
        }
}
return honorBiggestSum;
}
</pre>
```

Outstanding! You may take your time to refactor/comment your solution. Submit when ready.

Sample Tests

```
const chai = require("chai");
const assert = chai.assert;

describe("Fixed Tests", function() {
   it("regular", function() {
      assert.strictEqual(largestRadialSum([1,2,3,4], 2), 6);
      assert.strictEqual(largestRadialSum([1,5,6,3,4,2], 3), 11);
      assert.strictEqual(largestRadialSum([1,1,0], 1), 1);
});

it("edge cases", function() {
      assert.strictEqual(largestRadialSum([3], 1), 3);
      assert.strictEqual(largestRadialSum([9,10,2], 3), 21);
      assert.strictEqual(largestRadialSum([-2,-1,-2,-2], 2), -3);
});
```

> Random Testing

Random Testing

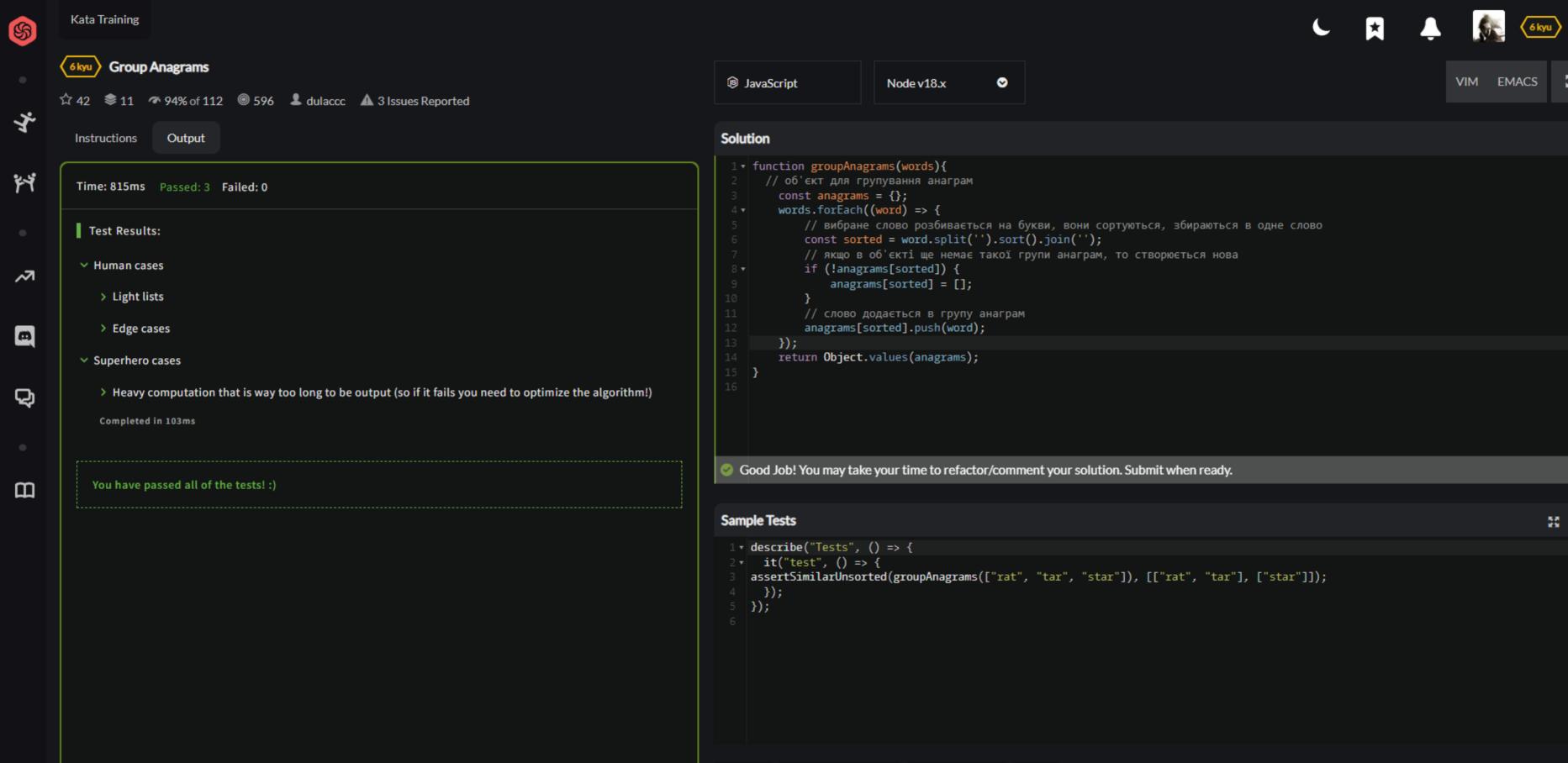
> Random Testing



```
Great! You may take your time to refactor/comment your solution. Submit when ready.
```

```
Sample Tests
```

```
const Test = require('@codewars/test-compat');
  describe("FindEvenIndex", function() {
  4 v it("Tests", function() {
        Test.assertEquals(findEvenIndex([1,2,3,4,3,2,1]),3, "The array was: [1,2,3,4,3,2,1] \n");
        Test.assertEquals(findEvenIndex([1,100,50,-51,1,1]),1, "The array was: [1,100,50,-51,1,1] \n");
        Test.assertEquals(findEvenIndex([1,2,3,4,5,6]),-1, "The array was: [1,2,3,4,5,6] \n");
        Test.assertEquals(findEvenIndex([20,10,30,10,10,15,35]),3, "The array was: [20,10,30,10,10,15,35] \n");
      });
10 });
```





VIM EMACS







33 3



🟠 19 比 🥏 7 🥟 92% of 107 🔘 164 of 420 👢 WellWellWell

> Random test 9

Instructions Output Time: 1024ms Passed: 105 Failed: 0 Test Results: Solution > only lays valid sausage packages > does not lay the 5th reward package > only lays valid sausage packages when only one box > lays no sausages when truck contains only other products > lays no sausages when truck is empty Completed in 4ms Random Tests > Random test 1 > Random test 2 > Random test 3 > Random test 4 > Random test 5 > Random test 6 > Random test 7 Random test 8

 JavaScript Node v18.x

Solution // розпаковка і перебір кожної пачки (кожного стрінга), якщо там сосиска і пачка не кратна 5, то вивести 2 * function unpackSausages(truck) { let packs = 0; let resultString = ""; truck.forEach((box) => { box.forEach((pack) => { if(unpackage(pack)){ packs++; if(packs % 5 !== 0){ const sausages = (pack.split('').slice(1, -1)); if(packs === 1){ resultString += sausages.join(' '); else{ resultString += " " + sausages.join(' '); })

Outstanding! You may take your time to refactor/comment your solution. Submit when ready.

②

Sample Tests

})

```
4 * describe("Solution", function() {
5 * it("only lays valid sausage packages", function() {
       assert.strictEqual(unpackSausages([["(----)", "[IIII]", "_HHH_"], ["IuI", "[))))]", "zz"], ["[@@@@]", "UwU", "[IIII]"]]), "I
    });
8 * it("does not lay the 5th reward package", function() {
       assert.strictEqual(unpackSausages([["[IIII]", "[llll]", "[1111]", "[@@@@]", "[|||||]", "[||||||]"]]), "I I I I l l l l l 1 1 1 1 @
    it("only lays valid sausage packages when only one box", function() {
       assert.strictEqual(unpackSausages([["[IIII]", "[||||||]", "[1-11]"]]), "I I I I | | | | | | ");
    it("lays no sausages when truck contains only other products", function() {
       assert.strictEqual(unpackSausages([[],[],["_"]]), "");
```

```
    JavaScript

                                                                                                                VIM EMACS
                                          ②
                 ❖
                         Node v8.1.3
Solution
   function getRootProperty(object, val) {
  if (object === val) {
             return true;
        через цикл нижче заходимо в рекурсію та одночасно він є перевіркою
         на те, чи є те, що передали в функцію об'єктом чи числом (якщо об'єкт, то
        функція ще раз рекурсивно викликається, якщо число, то Object.keys(obj)
         поверне пустий масив і цикл пропуститься, перейдемо відразу до return null
         бо це є числом, але не тим, яке нам потрібно, бо перевірка на початку
        obj === num повернула false
        for (const key of Object.keys(object)) {
            const recur = getRootProperty(object[key], val);
            if (recur) {
                 return key;
        return null;
Correct! You may take your time to refactor/comment your solution. Submit when ready.
Sample Tests
  describe("Tests", () => {
 2 * it("test", () => {
    //Basic test
 5 ▼ const object = {
        "one": {
             "nest1": {
```

"val1": [9, 34, 92, 100]

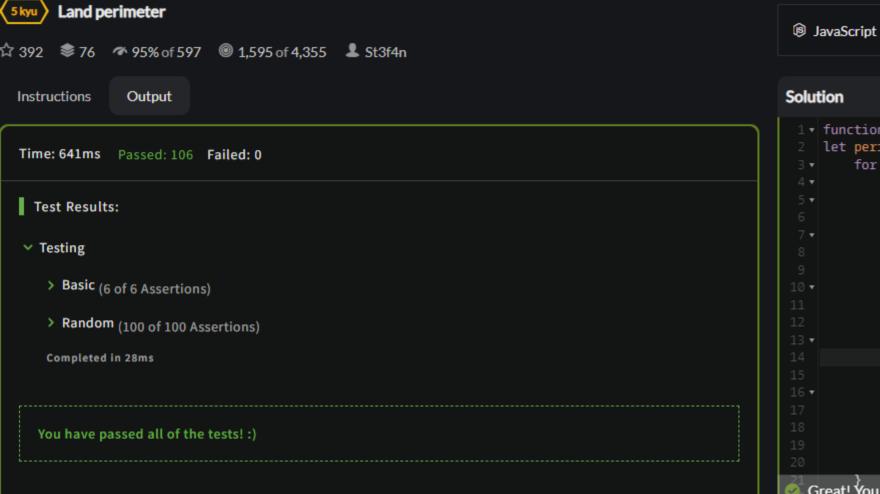
"n1": [10, 92, 53, 71], "n2": [82, 34, 6, 19]

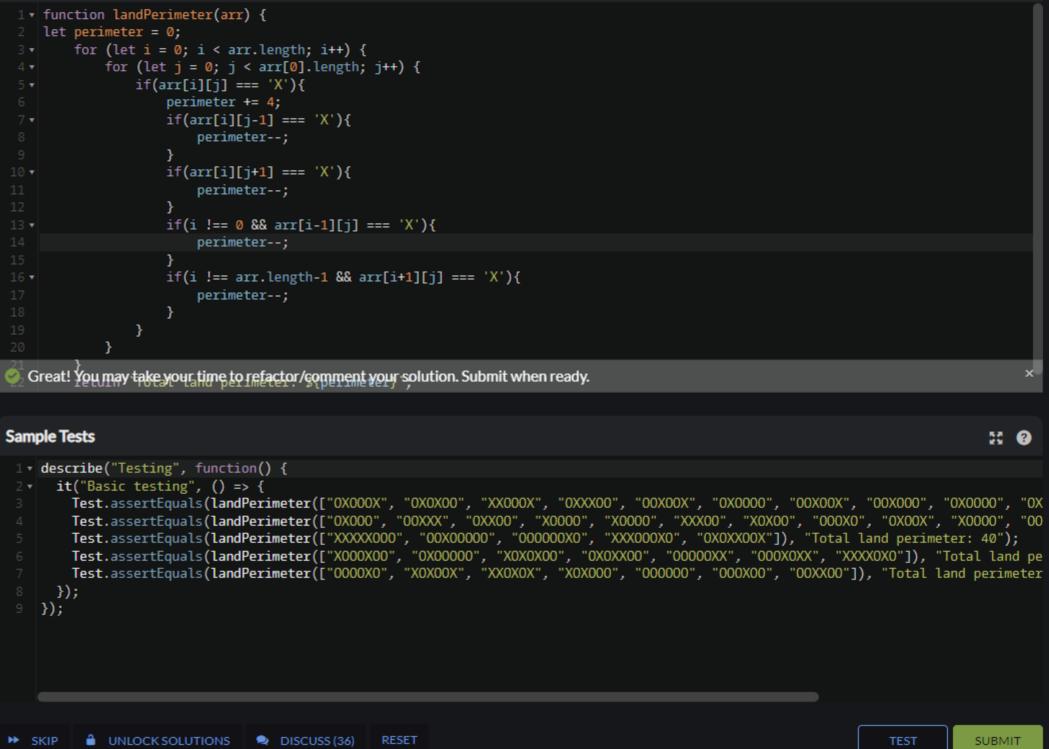
},

"2f7": {

33 9

Kata Iraining





0

Node v8.1.3

×

VIM EMACS