

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

Random Tests

random

Completed in 4ms

You have passed all of the tests! :)

JavaScript

Node v18.x

Solution

```
1 function largestRadialSum(arr, d) {
2   let honorSum;
3   let honorBiggestSum = Number.NEGATIVE_INFINITY;
4   for (let i = 0; i < arr.length/d; i++) {
5     honorSum = 0;
6     for (let j = 0; j < arr.length; j += arr.length/d) {
7       honorSum += arr[i + j];
8     }
9     if(honorBiggestSum < honorSum){
10      honorBiggestSum = honorSum;
11    }
12  }
13
14  return honorBiggestSum;
15 }
```

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

Sample Tests

```
1 const chai = require("chai");
2 const assert = chai.assert;
3
4 describe("Fixed Tests", function() {
5   it("regular", function() {
6     assert.strictEqual(largestRadialSum([1,2,3,4], 2), 6);
7     assert.strictEqual(largestRadialSum([1,5,6,3,4,2], 3), 11);
8     assert.strictEqual(largestRadialSum([1,1,0], 1), 1);
9   });
10  it("edge cases", function() {
11    assert.strictEqual(largestRadialSum([3], 1), 3);
12    assert.strictEqual(largestRadialSum([9,10,2], 3), 21);
13    assert.strictEqual(largestRadialSum([-2,-1,-2,-2], 2), -3);
14  });
15 }
```



6 kyu Equal Sides Of An Array

☆ 3199 📁 587 📈 91% of 11,981 🕒 32,598 of 109,994 👤 Shivo

Instructions

Output

Time: 866ms Passed: 51 Failed: 0

Test Results:

FindEvenIndex

Tests

Completed in 1ms

50 Random tests

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

Random Testing

JavaScript



Node v18.x



VIM

EMACS

Solution

```
1 function findEvenIndex(arr)
2 {
3   for (let i = 0; i < arr.length; i++) {
4     let leftSum = 0;
5     for (let j = 0; j < i; j++) {
6       leftSum += arr[j];
7     }
8     let rightSum = 0;
9     for (let j = i+1; j < arr.length; j++) {
10      rightSum += arr[j];
11    }
12    if(leftSum === rightSum){
13      return i;
14    }
15  }
16  return -1;
17 }
```

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

Sample Tests

```
1 const Test = require('@codewars/test-compat');
2
3 describe("FindEvenIndex", function() {
4   it("Tests", function() {
5     Test.assertEquals(findEvenIndex([1,2,3,4,3,2,1]),3, "The array was: [1,2,3,4,3,2,1] \n");
6     Test.assertEquals(findEvenIndex([1,100,50,-51,1,1]),1, "The array was: [1,100,50,-51,1,1] \n");
7     Test.assertEquals(findEvenIndex([1,2,3,4,5,6]),-1, "The array was: [1,2,3,4,5,6] \n");
8     Test.assertEquals(findEvenIndex([20,10,30,10,10,15,35]),3, "The array was: [20,10,30,10,10,15,35] \n");
9   });
10 });
```

Kata Training

6kyu

Group Anagrams

☆ 42

📄 11

📈 94% of 112

🕒 596

👤 dulaccc

⚠️ 3 Issues Reported

Instructions

Output

Time: 815ms

Passed: 3

Failed: 0

Test Results:

Human cases

> Light lists

> Edge cases

Superhero cases

> Heavy computation that is way too long to be output (so if it fails you need to optimize the algorithm!)

Completed in 103ms

You have passed all of the tests! :)

JavaScript

Node v18.x

VIM

EMACS

6kyu

Solution

```
1 function groupAnagrams(words){
2   // об'єкт для групування анаграм
3   const anagrams = {};
4   words.forEach((word) => {
5     // вибране слово розбивається на букви, вони сортуються, збираються в одне слово
6     const sorted = word.split('').sort().join('');
7     // якщо в об'єкті ще немає такої групи анаграм, то створюється нова
8     if (!anagrams[sorted]) {
9       anagrams[sorted] = [];
10    }
11    // слово додається в групу анаграм
12    anagrams[sorted].push(word);
13  });
14  return Object.values(anagrams);
15 }
16
```

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

Sample Tests

```
1 describe("Tests", () => {
2   it("test", () => {
3     assertSimilarUnsorted(groupAnagrams(["rat", "tar", "star"]), [["rat", "tar"], ["star"]]);
4   });
5 });
6
```

Kata Training

6 kyu

Unpack delicious sausages!

☆ 19

📚 7

📈 92% of 107

👁 164 of 420

👤 WellWellWell

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

> Random test 9

JavaScript

Node v18.x

VIMEMACS

Solution

```
1 // розпаковка і перебір кожної пачки (кожного стрінга), якщо там сосиска і пачка не кратна 5, то вивести
2 function unpackSausages(truck) {
3   let packs = 0;
4   let resultString = "";
5   truck.forEach((box) => {
6     box.forEach((pack) => {
7       if(unpackage(pack)){
8         packs++;
9         if(packs % 5 !== 0){
10           const sausages = (pack.split('').slice(1, -1));
11           if(packs === 1){
12             resultString += sausages.join(' ');
13           }
14           else{
15             resultString += " " + sausages.join(' ');
16           }
17         }
18       }
19     })
20   })
21   return resultString;
22 }
```

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

Sample Tests

```
3
4 describe("Solution", function() {
5   it("only lays valid sausage packages", function() {
6     assert.strictEqual(unpackSausages([["(---)", "[IIII]", "_HHH_"], ["IuI", "[])))]", "zz"], [["@@@@@", "UwU", "[I\\III]"]]), "I
7   });
8   it("does not lay the 5th reward package", function() {
9     assert.strictEqual(unpackSausages([["[IIII]", "[\\llll]", "[1111]", "@@@@@", "[\\llll]", "[\\llll\\ll]"]]), "I I I I \\llll\\ll 1 1 1 1 @
10  });
11  it("only lays valid sausage packages when only one box", function() {
12    assert.strictEqual(unpackSausages([["[IIII]", "[\\llll\\ll]", "[1-11]"]]), "I I I I \\llll\\ll");
13  });
14  it("lays no sausages when truck contains only other products", function() {
15    assert.strictEqual(unpackSausages([], [], ["_"]]), "");
16  });
17 }
```

Time: 5369ms

Passed: 600

Failed: 0

Test Results:

Random Test cases

- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == '3,5*y'
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == 'x.<ns'
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == 'r_v=d'
- ✔ Test Passed: Value == ')/5p%'
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == '5\'b4%'
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == null
- ✔ Test Passed: Value == '`kw-%'

JavaScript

Node v8.1.3

VIM

EMACS

Solution

```
1 function getRootProperty(object, val) {
2   if (object === val) {
3     return true;
4   }
5   /*
6    через цикл нижче заходимо в рекурсію та одночасно він є перевіркою
7    на те, чи є те, що передали в функцію об'єктом чи числом (якщо об'єкт, то
8    функція ще раз рекурсивно викликається, якщо число, то Object.keys(obj)
9    поверне пустий масив і цикл пропуститься, перейдемо відразу до return null
10   бо це є числом, але не тим, яке нам потрібно, бо перевірка на початку
11   obj === num повернула false
12   */
13   for (const key of Object.keys(object)) {
14     const recur = getRootProperty(object[key], val);
15     if (recur) {
16       return key;
17     }
18   }
19   return null;
20 }
```

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

Sample Tests

```
1 describe("Tests", () => {
2   it("test", () => {
3     //Basic test
4
5     const object = {
6       "one": {
7         "nest1": {
8           "val1": [9, 34, 92, 100]
9         }
10      },
11      "2f7": {
12        "n1": [10, 92, 53, 71],
13        "n2": [82, 34, 6, 19]
14      }
15    }
16  })
17 })
```

5 kyu Land perimeter

☆ 392 🏆 76 🔄 95% of 597 © 1,595 of 4,355 👤 St3f4n

Instructions

Output

Time: 641ms Passed: 106 Failed: 0

Test Results:

▼ Testing

➤ Basic (6 of 6 Assertions)

➤ Random (100 of 100 Assertions)

Completed in 28ms

You have passed all of the tests! :)

JavaScript

Node v8.1.3

VIM

EMACS

Solution

```
1 function landPerimeter(arr) {
2   let perimeter = 0;
3   for (let i = 0; i < arr.length; i++) {
4     for (let j = 0; j < arr[0].length; j++) {
5       if(arr[i][j] === 'X'){
6         perimeter += 4;
7         if(arr[i][j-1] === 'X'){
8           perimeter--;
9         }
10        if(arr[i][j+1] === 'X'){
11          perimeter--;
12        }
13        if(i !== 0 && arr[i-1][j] === 'X'){
14          perimeter--;
15        }
16        if(i !== arr.length-1 && arr[i+1][j] === 'X'){
17          perimeter--;
18        }
19      }
20    }
21  }
22  return "Total land perimeter: " + perimeter;
}
```

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

Sample Tests

```
1 describe("Testing", function() {
2   it("Basic testing", () => {
3     Test.assertEquals(landPerimeter(["0X000X", "0X0X00", "XX000X", "0XXX00", "00X00X", "0X0000", "00X00X", "00X000", "0X0000", "0X
4     Test.assertEquals(landPerimeter(["0X000", "00XXX", "0XX00", "X0000", "X0000", "XXX00", "X0X00", "000X0", "0X00X", "X0000", "00
5     Test.assertEquals(landPerimeter(["XXXXX000", "00X00000", "000000X0", "XXX000X0", "0X0XX00X"]), "Total land perimeter: 40");
6     Test.assertEquals(landPerimeter(["X000X00", "0X00000", "X0X0X00", "0X0XX00", "00000XX", "000X0XX", "XXXX0X0"]), "Total land pe
7     Test.assertEquals(landPerimeter(["0000X0", "X0X00X", "XX0X0X", "X0X000", "000000", "000X00", "00XX00"]), "Total land perimeter
8   });
9 });
```

SKIP

UNLOCK SOLUTIONS

DISCUSS (36)

RESET

TEST

SUBMIT