You can view this report online at : https://www.hackerrank.com/x/tests/969534/candidates/20004993/report

Full Name:	Omar A Hernandez
Email:	oah232@gmail.com
Test Name:	**App Academy HackerRank Challenge 12/14**
Taken On:	16 Dec 2020 10:11:45 PST
Time Taken:	73 min 3 sec/ 105 min
Work Experience:	< 1 years
Invited by:	Sami Ellaboudy
Invited on:	14 Dec 2020 15:22:35 PST
Skills Score:	Problem Solving (Basic) 93/100 Problem Solving (Intermediate) 75/75
Tags Score:	Algorithms 168/175 Arrays 168/175 Binary Search 75/75 CSS3 2.5/5 Data Structures 118/125 Easy 95.5/105 HTTP 2.5/5 Implementation 50/50 Interviewer Guidelines 75/75 Medium 80/85 MongoDB 2.5/5 Problem Solving 168/175 REST API 2.5/5 Theme: Finance 75/75

92.4%

175/190

scored in **App Academy HackerRank Challenge 12/14** in 73 min 3 sec on 16 Dec 2020 10:11:45 PST

Recruiter/Team Comments:

No Comments.

	Question Description	Time Taken	Score	Status
Q1	Grouping Digits > **Coding**	37 min 11 sec	43/ 50	✓
Q2	Profit Targets > **Coding**	22 min 52 sec	75/ 75	✓
Q3	Game Winner > **Coding**	10 min 35 sec	50/ 50	✓
Q4	HTTP status codes > **Multiple Choice**	26 sec	2.5/ 5	✓
Q5	MongoDB: Delete > **Multiple Choice**	12 sec	2.5/ 5	✓

Q6	Selector > Multiple Choice	1 min 37 sec	2.5/ 5	✓

QUESTION 1

✓

Correct Answer

Score 43

Grouping Digits › Coding

Data Structures | Easy | Algorithms | Arrays | Problem Solving

QUESTION DESCRIPTION

Given an array of binary digits, *0* and *1*, sort the array so that all zeros are at one end and all ones are at the other. Which end does not matter. To sort the array, swap any two adjacent elements. Determine the minimum number of swaps to sort the array.

Example

arr=[0,1,0,1]

With *1* move, switching elements *1* and *2*, yields *[0,0,1,1]*, a sorted array.

Function Description

Complete the function *minMoves* in the editor below.

minMoves has the following parameter(s):

int arr[n]: an array of binary digits

Returns

int: the minimum number of moves necessary

Constraints

- $1 \le n \le 10^5$
- *arr[i] is in the set {0,1}*

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array *arr[]*.

Each of the next *n* lines contains a binary digit as an integer, *arr[i]*.

▼ Sample Case 0

Sample Input

```
STDIN      Function
-----      -----
8       →  arr[i] size n = 8
1       →  arr = [1, 1, 1, 1, 0, 0, 0, 0]
1
1
1
0
0
0
0
```

Sample Output

```
0
```

Explanation

The array is already sorted, so no moves are necessary.

▼ Sample Case 1

/

▼ Sample Case 1

Sample Input

```
STDIN     Function
-----     -----
8       →  arr[i] size n = 8
1       →  arr = [1, 1, 1, 1, 0, 1, 0, 1]
1
1
1
0
1
0
1
```

Sample Output

```
3
```

Explanation

Perform the following minimal sequence of *3* moves `1 1 1 1 0 1 0 1` → `1 1 1 1` **`1`** `0 0 1` → `1 1 1 1 1 0` **`1`** `0` → `1 1 1 1 1` **`1`** `0 0` to sort the array. Bold/red is the value at the new position.

▼ Sample Case 2

Sample Input

```
STDIN     Function
-----     -----
8       →  arr[i] size n = 8
1       →  arr = [1, 0, 1, 0, 0, 0, 0, 1]
0
1
0
0
0
0
1
```

Sample Output

```
6
```

Explanation

This is the minimal sequence of *6* moves `1 0 1 0 0 0 0 1` → `1` **`1`** `0 0 0 0 0 1` → `1 1 0 0 0 0` **`1`** `0` → `1 1 0 0 0` **`1`** `0 0` → `1 1 0 0` **`1`** `0 0 0` → `1 1 0` **`1`** `0 0 0 0` → `1 1` **`1`** `0 0 0 0 0`.

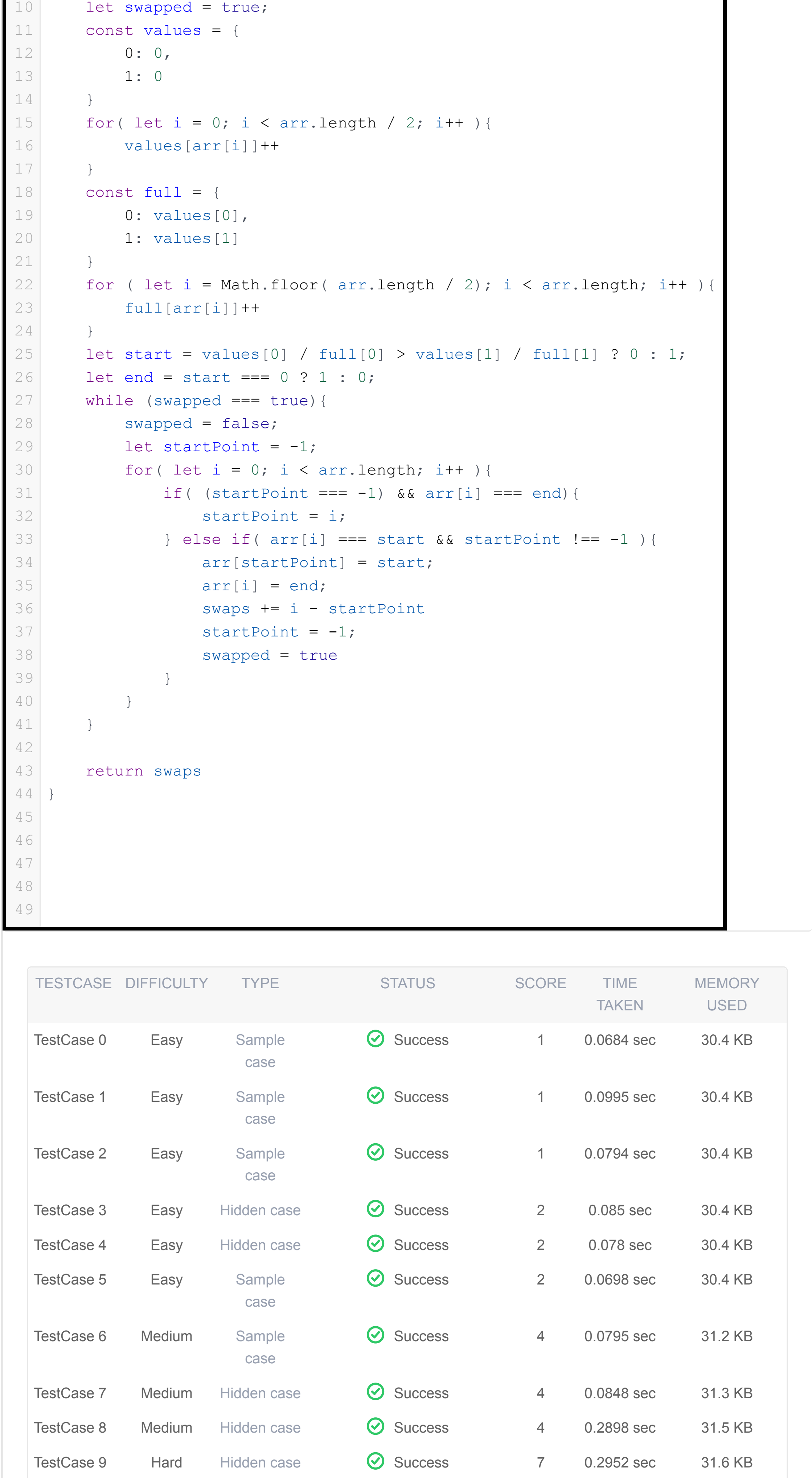
CANDIDATE ANSWER

Language used: **JavaScript (Node.js)**

```javascript
/*
 * Complete the 'minMoves' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

function minMoves(arr) {
    let swaps = 0;
```

/



```
    let swapped = true;
    const values = {
        0: 0,
        1: 0
    }
    for( let i = 0; i < arr.length / 2; i++ ){
        values[arr[i]]++
    }
    const full = {
        0: values[0],
        1: values[1]
    }
    for ( let i = Math.floor( arr.length / 2); i < arr.length; i++ ){
        full[arr[i]]++
    }
    let start = values[0] / full[0] > values[1] / full[1] ? 0 : 1;
    let end = start === 0 ? 1 : 0;
    while (swapped === true){
        swapped = false;
        let startPoint = -1;
        for( let i = 0; i < arr.length; i++ ){
            if( (startPoint === -1) && arr[i] === end){
                startPoint = i;
            } else if( arr[i] === start && startPoint !== -1 ){
                arr[startPoint] = start;
                arr[i] = end;
                swaps += i - startPoint
                startPoint = -1;
                swapped = true
            }
        }
    }

    return swaps
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	✓ Success	1	0.0684 sec	30.4 KB
TestCase 1	Easy	Sample case	✓ Success	1	0.0995 sec	30.4 KB
TestCase 2	Easy	Sample case	✓ Success	1	0.0794 sec	30.4 KB
TestCase 3	Easy	Hidden case	✓ Success	2	0.085 sec	30.4 KB
TestCase 4	Easy	Hidden case	✓ Success	2	0.078 sec	30.4 KB
TestCase 5	Easy	Sample case	✓ Success	2	0.0698 sec	30.4 KB
TestCase 6	Medium	Sample case	✓ Success	4	0.0795 sec	31.2 KB
TestCase 7	Medium	Hidden case	✓ Success	4	0.0848 sec	31.3 KB
TestCase 8	Medium	Hidden case	✓ Success	4	0.2898 sec	31.5 KB
TestCase 9	Hard	Hidden case	✓ Success	7	0.2952 sec	31.6 KB

/

TestCase 10	Hard	Hidden case	⊗ Terminated due to timeout	0	10.0042 sec	35.7 KB
TestCase 11	Hard	Hidden case	✓ Success	7	0.0984 sec	35.4 KB
TestCase 12	Hard	Hidden case	✓ Success	8	0.0979 sec	35.5 KB

No Comments

QUESTION 2

✓ Correct Answer

Score 75

Profit Targets › Coding

Binary Search | Data Structures | Medium | Algorithms | Arrays | Problem Solving | Theme: Finance | Interviewer Guidelines

QUESTION DESCRIPTION

A financial analyst is responsible for a portfolio of profitable stocks represented in an array. Each item in the array represents the yearly profit of a corresponding stock. The analyst gathers all distinct pairs of stocks that reached the target profit. Distinct pairs are pairs that differ in at least one element. Given the array of profits, find the number of distinct pairs of stocks where the sum of each pair's profits is exactly equal to the target profit.

Example

stocksProfit = [5, 7, 9, 13, 11, 6, 6, 3, 3]

target = 12 profit's target

- There are *4* pairs of stocks that have the sum of their profits equals to the target 12 . Note that because there are two instances of *3* in *stocksProfit* there are two pairs matching (9, 3): *stocksProfits* indices 2 and 7, and indices 2 and 8, but only one can be included.
- There are *3* distinct pairs of stocks: *(5, 7), (3, 9),* and *(6, 6)* and the return value is *3*.

Function Description

Complete the function *stockPairs* in the editor below.

stockPairs has the following parameter(s):

int stocksProfit[n]: an array of integers representing the stocks profits

target: an integer representing the yearly target profit

Returns:

int: the total number of pairs determined

Constraints

- $1 \le n \le 5 \times 10^5$
- $0 \le stocksProfit[i] \le 10^9$
- $0 \le target \le 5 \times 10^9$

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array *stocksProfit*.

The next *n* lines each contain an element *stocksProfit[i]* where $0 \le i < n$.

The next line contains an integer *target*, the target value.

▼ Sample Case 0

Sample Input 0

```
STDIN       Function
-----       --------
6       →   stocksProfit[] size n = 6
1       →   stocksProfit = [1, 3, 46, 1, 3, 9]
3
46
1
3
9
47      →   target = 47
```

Sample Output 0

```
1
```

Explanation 0

There are *4* pairs where *stocksProfit[i]* + *stocksProfit[j]* = 47

1. *(stocksProfit0] = 1, stocksProfit[2] = 46)*
2. *(stocksProfit[2] = 46, stocksProfit[0] = 1)*
3. *(stocksProfit[2] = 46, stocksProfit[3] = 1)*
4. *(stocksProfit[3] = 1, stocksProfit[2] = 46)*

Since all four pairs contain the same values, there is only *1 distinct* pair of stocks : *(1, 46)*.

▼ Sample Case 1

Sample Input 1

```
STDIN       Function
-----       --------
7       →   stocksProfit[] size n = 7
6       →   stocksProfit = [6, 6, 3, 9, 3, 5, 1]
6
3
9
3
5
1
12      →   target = 12
```

Sample Output 1

```
2
```

Explanation 1

There are *5* pairs where *stocksProfit[i]* + *stocksProfit[j]* = 12:

1. *(stocksProfit[0] = 6, stocksProfit[1] = 6)*
2. *(stocksProfit[1] = 6, stocksProfit[0] = 6)*
3. *(stocksProfit[2] = 3, stocksProfit[3] = 9)*
4. *(stocksProfit[3] = 9, stocksProfit[2] = 3)*
5. *(stocksProfit[3] = 9, stocksProfit[4] = 3)*
6. *(stocksProfit[4] = 3, stocksProfit[3] = 9)*

The first *2* pairs are the same, as are the last *4.* There are only *2 distinct* pairs of stocks: *(3, 9)* and *(6, 6)*.

INTERVIEWER GUIDELINES

▼ Hint 1

Is there an efficient way you can find out whether target - stocksProfit[i] exists in the array for every i?

▼ Hint 2

Multiple occurrences of the same value don't contribute to the final answer except in one special case, target/2 when target is even. Try using hash tables.

▼ Solution

/

▼ Solution

Concepts covered: Hash Table

Optimal Solution:

Suppose that we already know the value of the first stock, call it *value*. We can say that the value of the second stock must be target - value. Then we just need to find out whether target - value exists in the array. We can to this efficiently using a hash table. One point to notice here is that if target is divisible by 2, then there must be at least two occurrences of target/2 in the array for it to contribute in the final answer.

```
def stockPairs(stocksProfit, target):
    stock_values = set(stocksProfit)
    ans = 0
    for value in stock_values:
        if target - value in stock_values and target != 2 * value:
            ans += 1
    if target % 2 == 0 and stocksProfit.count(target // 2) > 1:
        ans += 2
    return ans // 2
```

Brute Force Approach: Passes 13 of 15 test cases

```
def stockPairs(stocksProfit, target):
    values_taken = set()
    ans = 0
    n = len(stocksProfit)
    for i in range(n):
        for j in range(i+1, n):
            if stocksProfit[i] + stocksProfit[j] == target and
(min(stocksProfit[i], stocksProfit[j]), max(stocksProfit[i],
stocksProfit[j])) not in values_taken:
                ans += 1
                values_taken.add((min(stocksProfit[i], stocksProfit[j]),
max(stocksProfit[i], stocksProfit[j])))
    return ans
```

Error Handling: The edge case which candidates must take care is when target is divisible by 2 and the number of occurrences of target/2 is equal to 1.

▼ Complexity Analysis

Time Complexity - O(n).

Since we are iterating over each element exactly once and for each element we are doing a lookup in the hash table (O(1) time complexity), each pass costs O(1) time. The overall time complexity is O(n).

Space Complexity - O(n).

The hash table takes O(n) space.

▼ Follow up Question

What if the task is to find out the number of distinct pair of stocks such that their sum is ≥ target?

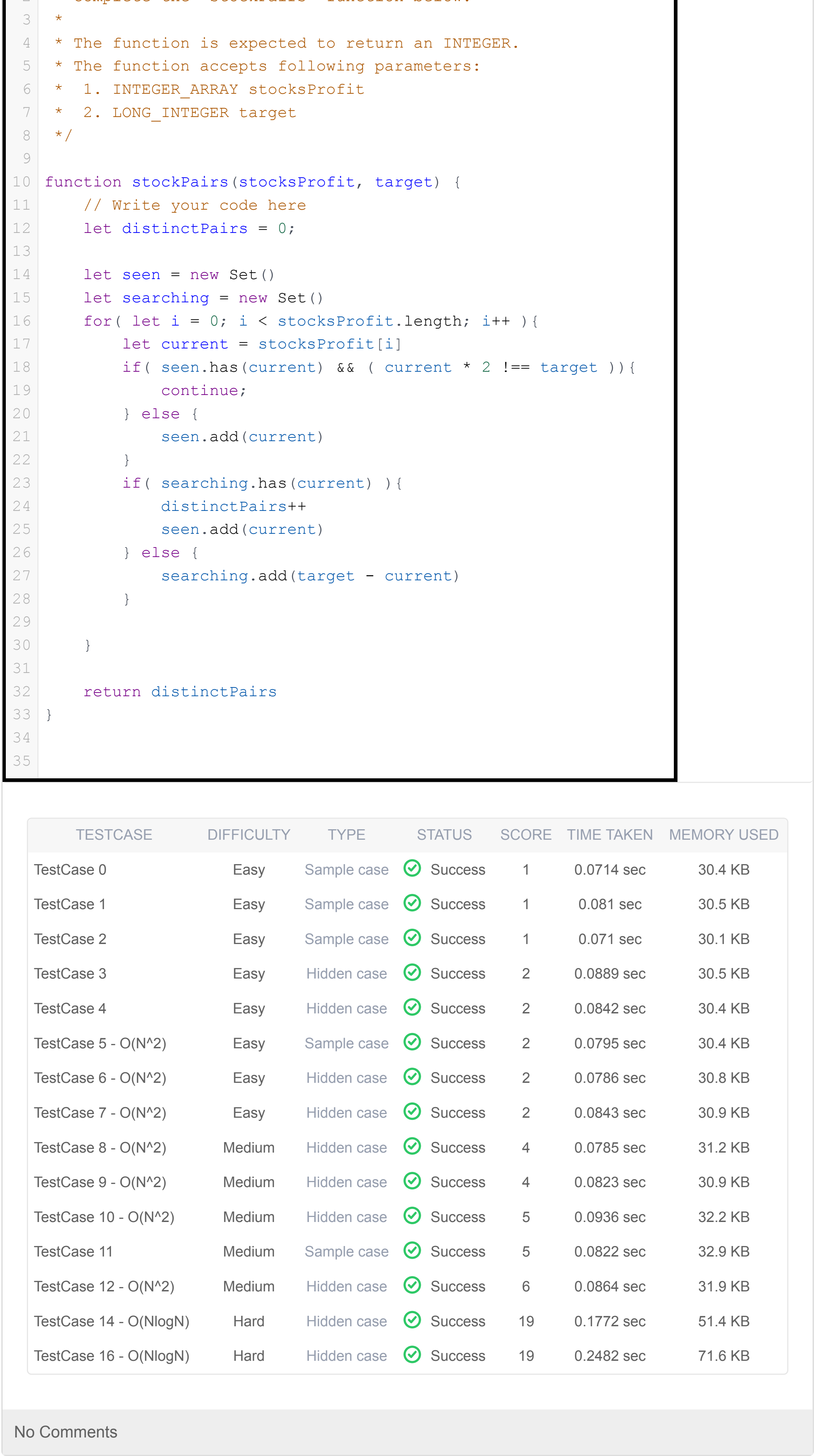
Now, for each element value we need to query the number of integers which are ≥ target - value. This can be done using a binary search tree.

CANDIDATE ANSWER

Language used: **JavaScript (Node.js)**

```
/*
 * Complete the 'stockPairs' function below.
```

/



```
 * Complete the 'stockPairs' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts following parameters:
 *  1. INTEGER_ARRAY stocksProfit
 *  2. LONG_INTEGER target
 */

function stockPairs(stocksProfit, target) {
    // Write your code here
    let distinctPairs = 0;

    let seen = new Set()
    let searching = new Set()
    for( let i = 0; i < stocksProfit.length; i++ ){
        let current = stocksProfit[i]
        if( seen.has(current) && ( current * 2 !== target )){
            continue;
        } else {
            seen.add(current)
        }
        if( searching.has(current) ){
            distinctPairs++
            seen.add(current)
        } else {
            searching.add(target - current)
        }

    }

    return distinctPairs
}
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	✓ Success	1	0.0714 sec	30.4 KB
TestCase 1	Easy	Sample case	✓ Success	1	0.081 sec	30.5 KB
TestCase 2	Easy	Sample case	✓ Success	1	0.071 sec	30.1 KB
TestCase 3	Easy	Hidden case	✓ Success	2	0.0889 sec	30.5 KB
TestCase 4	Easy	Hidden case	✓ Success	2	0.0842 sec	30.4 KB
TestCase 5 - O(N^2)	Easy	Sample case	✓ Success	2	0.0795 sec	30.4 KB
TestCase 6 - O(N^2)	Easy	Hidden case	✓ Success	2	0.0786 sec	30.8 KB
TestCase 7 - O(N^2)	Easy	Hidden case	✓ Success	2	0.0843 sec	30.9 KB
TestCase 8 - O(N^2)	Medium	Hidden case	✓ Success	4	0.0785 sec	31.2 KB
TestCase 9 - O(N^2)	Medium	Hidden case	✓ Success	4	0.0823 sec	30.9 KB
TestCase 10 - O(N^2)	Medium	Hidden case	✓ Success	5	0.0936 sec	32.2 KB
TestCase 11	Medium	Sample case	✓ Success	5	0.0822 sec	32.9 KB
TestCase 12 - O(N^2)	Medium	Hidden case	✓ Success	6	0.0864 sec	31.9 KB
TestCase 14 - O(NlogN)	Hard	Hidden case	✓ Success	19	0.1772 sec	51.4 KB
TestCase 16 - O(NlogN)	Hard	Hidden case	✓ Success	19	0.2482 sec	71.6 KB

No Comments

/

QUESTION 3

Correct Answer

Score 50

Game Winner › Coding

Implementation | Problem Solving | Easy | Algorithms | Arrays

QUESTION DESCRIPTION

Two players are playing a game where white or black pieces are represented by a string, *colors*. The game rules are as follows:

- *Wendy* moves first and then they take alternate turns.
- With each move, *Wendy* may remove a white piece that has adjacent white pieces on both sides.
- Likewise, with each move, *Bob* may remove any black piece that has adjacent black pieces on both sides.
- After a piece is removed, the string is reduced in size by one piece. For instance, removing '*Y'* from '*XYZ'* results in '*XZ'*.
- When a player can no longer move, they have lost the game.

Example

colors = 'wwwbbbbwww'

Wendy removes the piece *w* at index *1*, *colors = 'wwbbbbwww'*

Bob removes the piece *b* at index *3*, *colors = 'wwbbbwww'*

Wendy removes the piece *w* at index *6*, *colors = 'wwbbbww'*

Bob removes the piece *b* from index *3*, *colors = 'wwbbww'*

Wendy has no other move, so *Bob* wins. Return *'bob'*.

Determine who wins if *Wendy* and *Bob* both play with optimum skill. Return the string *'wendy'* or *'bob'*.

Function Description

Complete the function g*ameWinner* in the editor below.

gameWinner has the following parameter(s):

string *colors*: each *colors[i]* represents the color located at the index of *i* within the string

Returns;

string: the winner of the game, either '*wendy'* or *'bob'*.

Constraints

- *colors[i] is either 'w' or 'b'*
- *1* ≤ size of *colors* ≤ 10^6

▼ Input Format Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first and only line of input contains a single string *colors*.

▼ Sample Case 0

Sample Input

```
STDIN      Function
-----      -----
wwwbb   →  colors = 'wwwbb'
```

Sample Output

```
wendy
```

Explanation

/

There are five colors in the string. Wendy can remove any of the white pieces in the first move. After that, the *colors* string becomes '*wwbb'*. Bob has no move since there is no black piece with exactly two *black* adjacent pieces, so Wendy wins.

CANDIDATE ANSWER

Language used: **JavaScript (Node.js)**

```
/*
 * Complete the 'gameWinner' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts STRING colors as parameter.
 */

function gameWinner(colors) {
    // Write your code here
    let sections = [];
    let currentColor = colors[0]
    let startPos = 0
    for( let i = 1; i <= colors.length; i++ ){
        let color = colors[i]
        if( color !== currentColor ){
            sections.push(colors.slice(startPos, i));
            startPos = i;
            currentColor = color
        }
    }

    let moves = { 'w': 0, 'b': 0}

    sections.forEach( str => {
        if ( str.length >= 3 ){
            moves[str[0]] += str.length
        }
    })
    return moves['w'] > moves['b'] ? "wendy" : "bob"
}


```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
TestCase 0	Easy	Sample case	✓ Success	1	0.0728 sec	30.5 KB
TestCase 1	Easy	Sample case	✓ Success	1	0.0708 sec	30.4 KB
TestCase 2	Easy	Sample case	✓ Success	1	0.0805 sec	30.4 KB
TestCase 3	Easy	Sample case	✓ Success	5	0.0867 sec	30.5 KB
TestCase 4	Easy	Sample case	✓ Success	6	0.0711 sec	30.5 KB
TestCase 5	Medium	Hidden case	✓ Success	4	0.0739 sec	30.4 KB
TestCase 6	Medium	Hidden case	✓ Success	4	0.2268 sec	56.8 KB
TestCase 7	Medium	Hidden case	✓ Success	4	0.1032 sec	34 KB
TestCase 8	Medium	Hidden case	✓ Success	4	0.096 sec	34 KB
TestCase 9	Hard	Hidden case	✓ Success	5	0.1761 sec	65.3 KB
TestCase 10	Hard	Hidden case	✓ Success	5	0.1293 sec	38.2 KB

/

TestCase 11	Hard	Hidden case	✓ Success	5	0.1769 sec	65.1 KB
TestCase 12	Hard	Hidden case	✓ Success	5	0.1946 sec	53.5 KB

No Comments

QUESTION 4

✓

Correct Answer

Score 2.5

HTTP status codes › Multiple Choice

Medium | REST API | HTTP

QUESTION DESCRIPTION

HTTP uses status codes to indicate how a specific HTTP request was handled. Which of the following are true?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ● HTTP status codes are divided into 4 main groups based on the kind of response they are related to.
- ✓ ○ HTTP status codes are divided into 5 main groups based on the kind of response they are related to.
- ✓ ● In case of an unauthorized access attempt, the HTTP status code 401 should be returned. If access is authorized but the client is lacking required permissions, the HTTP status code 403 should be returned.
- ○ In case of an unauthorized access attempt, HTTP status code 403 should be returned. If access is authorized but the client is lacking required permissions, the HTTP status code 401 should be returned

No Comments

QUESTION 5

✓

Correct Answer

Score 2.5

MongoDB: Delete › Multiple Choice

MongoDB | Easy

QUESTION DESCRIPTION

Which of the following queries will remove only the first matching document from the transactions collection?

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ ○ db.transactions.remove({}, true)
- ○ db.transactions.deleteMany({})
- ○ db.transactions.delete({
- ✓ ● db.transactions.deleteOne({})

No Comments

/

QUESTION 6

✓ Correct Answer

Score 2.5

Selector › Multiple Choice Medium CSS3

QUESTION DESCRIPTION

Given the style below, which of the following templates will produce '<ul>' with 'red' background?

```
<style>
p ~ ul {
  background: red;
}
</style>
```

INTERVIEWER GUIDELINES

Understanding on CSS selectors

CANDIDATE ANSWER

Options: (Expected answer indicated with a tick)

- ✓ ○ <p> <ul> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul> </p>
- ✓ ● <p></p> <ul> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul>
- ○ <ul> <li>Coffee</li> <li>Tea</li> <li>Milk</li> </ul> <p></p>
- ○ All the above

No Comments

PDF generated at: 16 Dec 2020 19:26:22 UTC