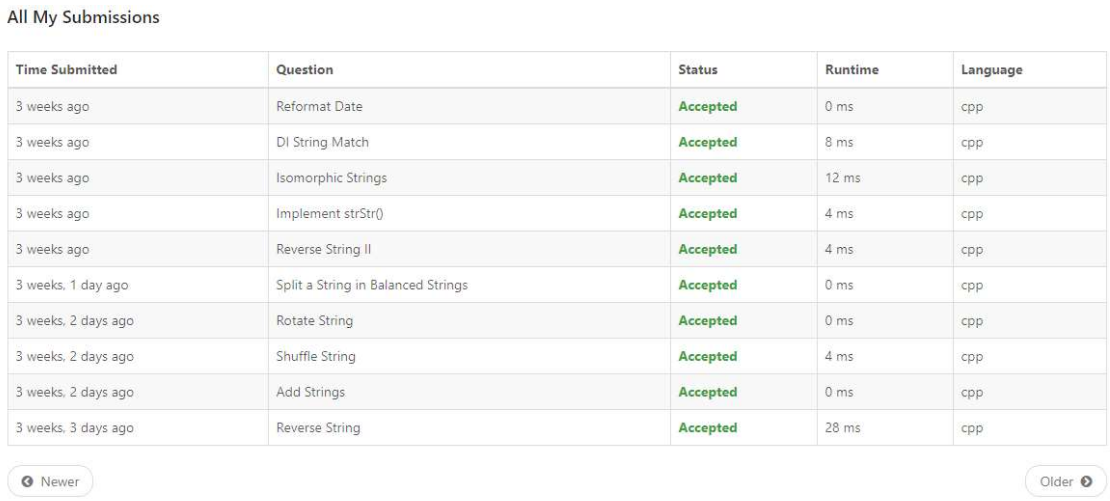
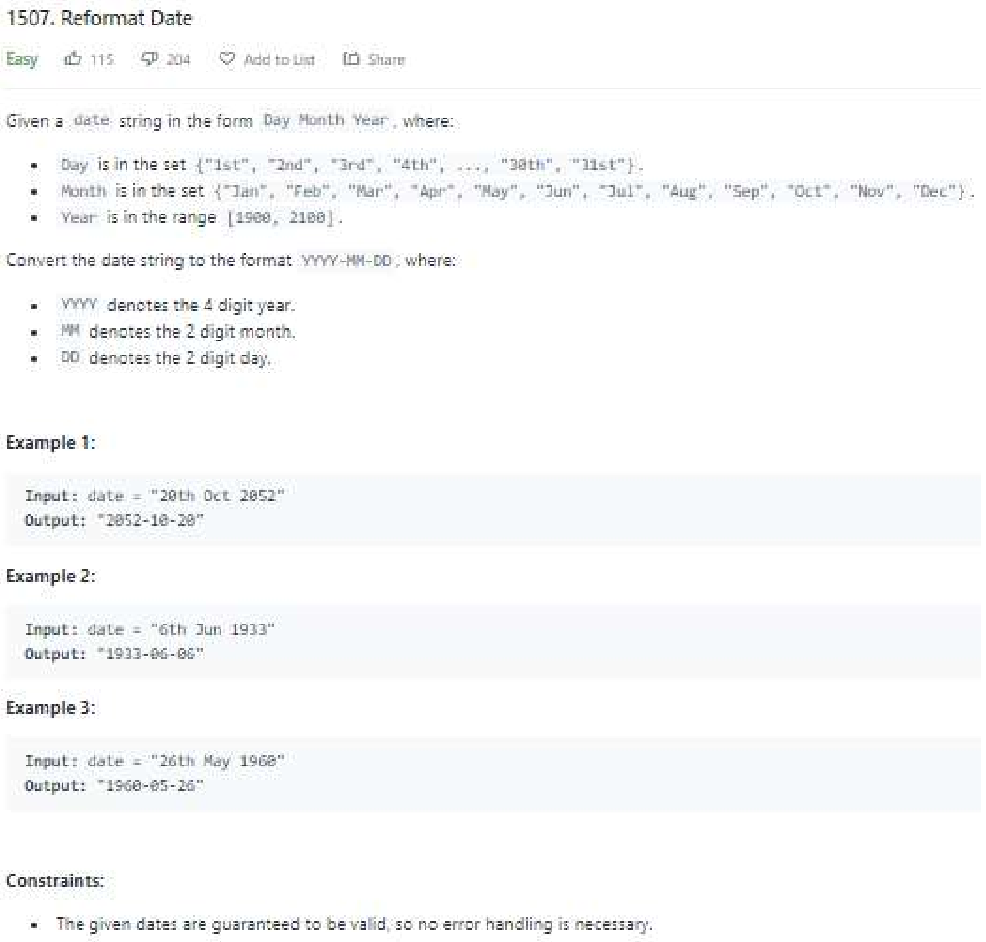
Submission Summary



All My Submissions

Time Submitted	Question	Status	Runtime	Language
3 weeks ago	Reformat Date	Accepted	0 ms	cpp
3 weeks ago	DI String Match	Accepted	8 ms	cpp
3 weeks ago	Isomorphic Strings	Accepted	12 ms	cpp
3 weeks ago	Implement strStr()	Accepted	4 ms	cpp
3 weeks ago	Reverse String II	Accepted	4 ms	cpp
3 weeks, 1 day ago	Split a String in Balanced Strings	Accepted	0 ms	cpp
3 weeks, 2 days ago	Rotate String	Accepted	0 ms	cpp
3 weeks, 2 days ago	Shuffle String	Accepted	4 ms	cpp
3 weeks, 2 days ago	Add Strings	Accepted	0 ms	cpp
3 weeks, 3 days ago	Reverse String	Accepted	28 ms	cpp

Question 1.



1507. Reformat Date

Given a `date` string in the form `Day Month Year`, where:

- `Day` is in the set `{"1st", "2nd", "3rd", "4th", ..., "30th", "31st"}`.
- `Month` is in the set `{"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"}`.
- `Year` is in the range `[1900, 2100]`.

Convert the date string to the format `YYYY-MM-DD`, where:

- `YYYY` denotes the 4 digit year.
- `MM` denotes the 2 digit month.
- `DD` denotes the 2 digit day.

Example 1:

```
Input: date = "20th Oct 2052"
Output: "2052-10-20"
```

Example 2:

```
Input: date = "6th Jun 1933"
Output: "1933-06-06"
```

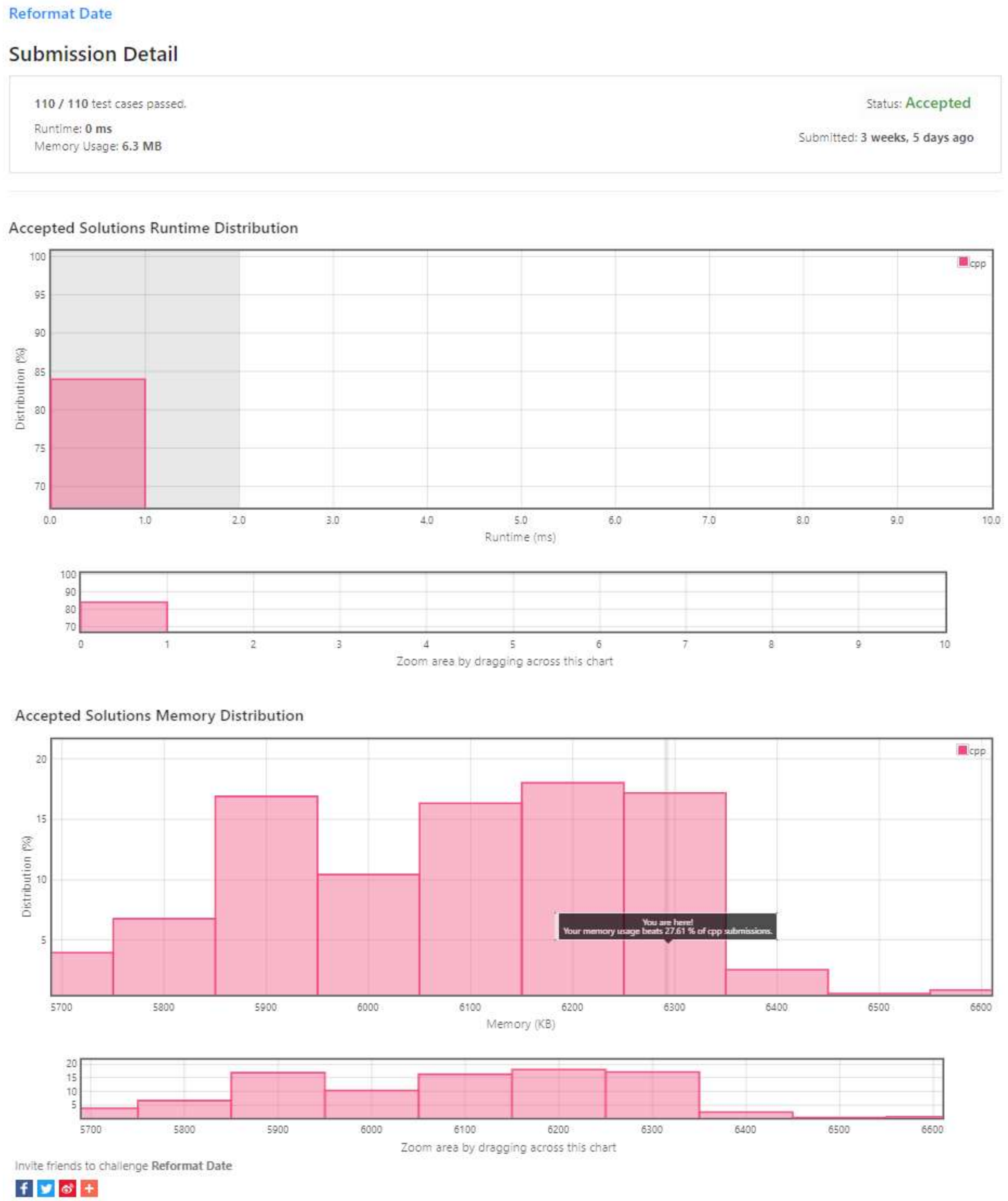
Example 3:

```
Input: date = "26th May 1960"
Output: "1960-05-26"
```

Constraints:

- The given dates are guaranteed to be valid, so no error handling is necessary.

Solution 1:



Reformat Date

Submission Detail

110 / 110 test cases passed.

Status: **Accepted**

Runtime: **0 ms**

Memory Usage: **6.3 MB**

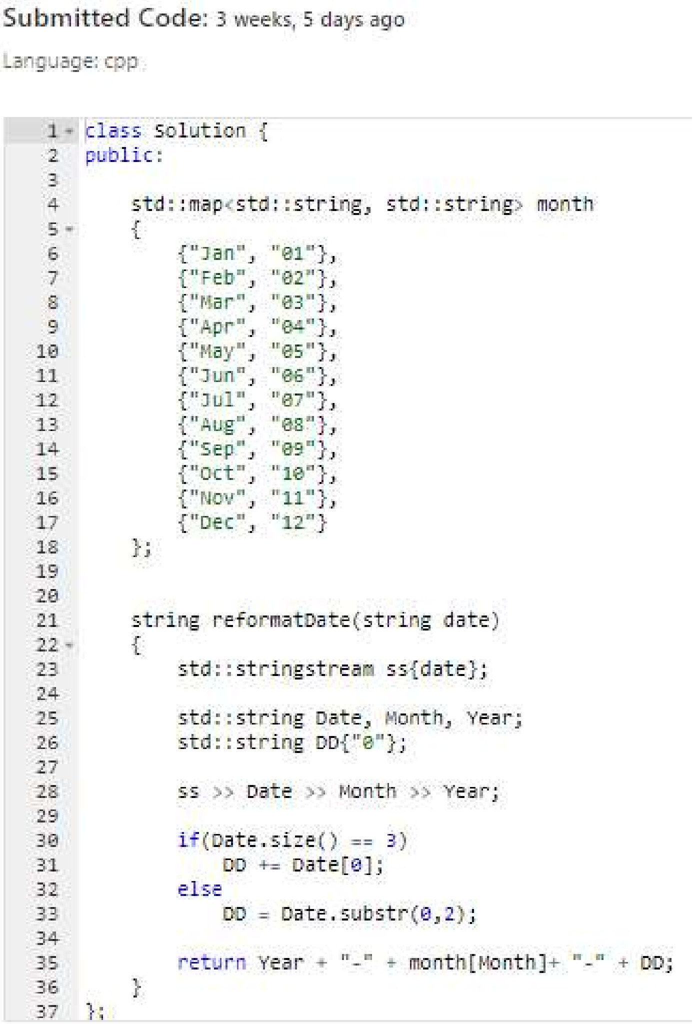
Submitted: **3 weeks, 5 days ago**

Accepted Solutions Runtime Distribution

Accepted Solutions Memory Distribution

You are here!
Your memory usage beats 27.61 % of cpp submissions.

Invite friends to challenge **Reformat Date**

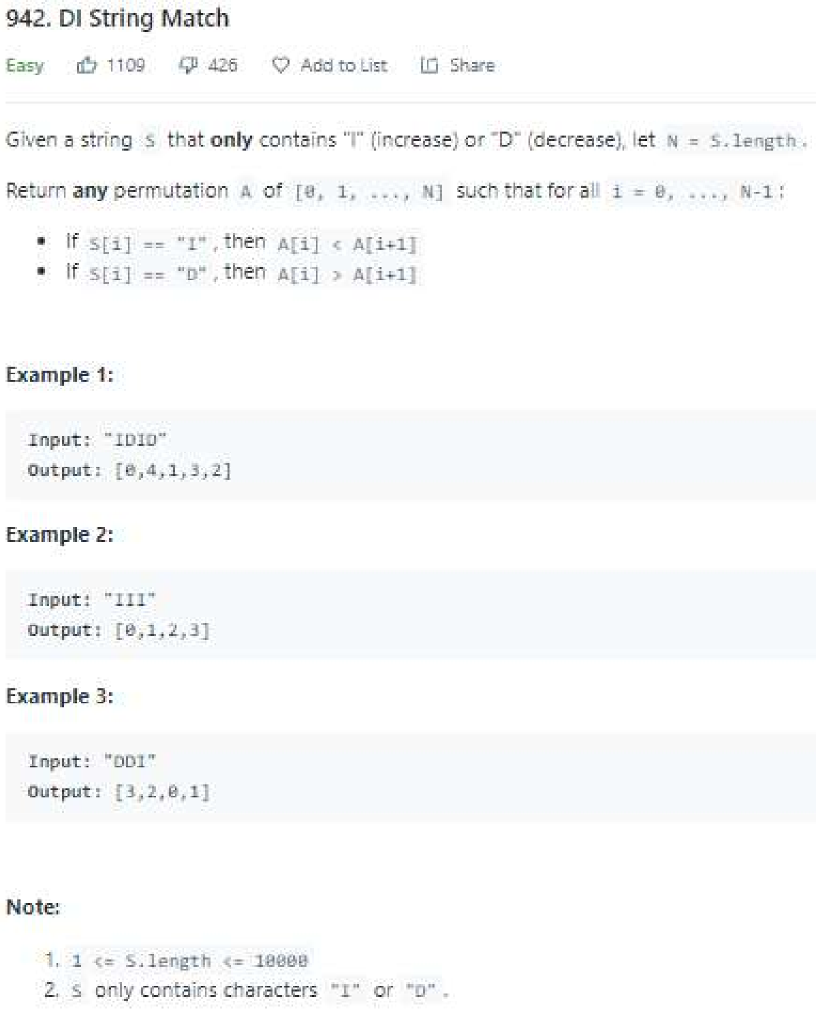


Submitted Code: 3 weeks, 5 days ago

Language: cpp

```cpp
class Solution {
public:

    std::map<std::string, std::string> month
    {
        {"Jan", "01"},
        {"Feb", "02"},
        {"Mar", "03"},
        {"Apr", "04"},
        {"May", "05"},
        {"Jun", "06"},
        {"Jul", "07"},
        {"Aug", "08"},
        {"Sep", "09"},
        {"Oct", "10"},
        {"Nov", "11"},
        {"Dec", "12"}
    };


    string reformatDate(string date)
    {
        std::stringstream ss{date};

        std::string Date, Month, Year;
        std::string DD{"0"};

        ss >> Date >> Month >> Year;

        if(Date.size() == 3)
            DD += Date[0];
        else
            DD = Date.substr(0,2);

        return Year + "-" + month[Month]+ "-" + DD;
    }
};
```

Question 2:



942. DI String Match

Given a string `S` that **only** contains "I" (increase) or "D" (decrease), let `N = S.length`.

Return **any** permutation `A` of `[0, 1, ..., N]` such that for all `i = 0, ..., N-1`:

- If `S[i] == "I"`, then `A[i] < A[i+1]`
- If `S[i] == "D"`, then `A[i] > A[i+1]`

Example 1:

```
Input: "IDID"
Output: [0,4,1,3,2]
```

Example 2:

```
Input: "III"
Output: [0,1,2,3]
```

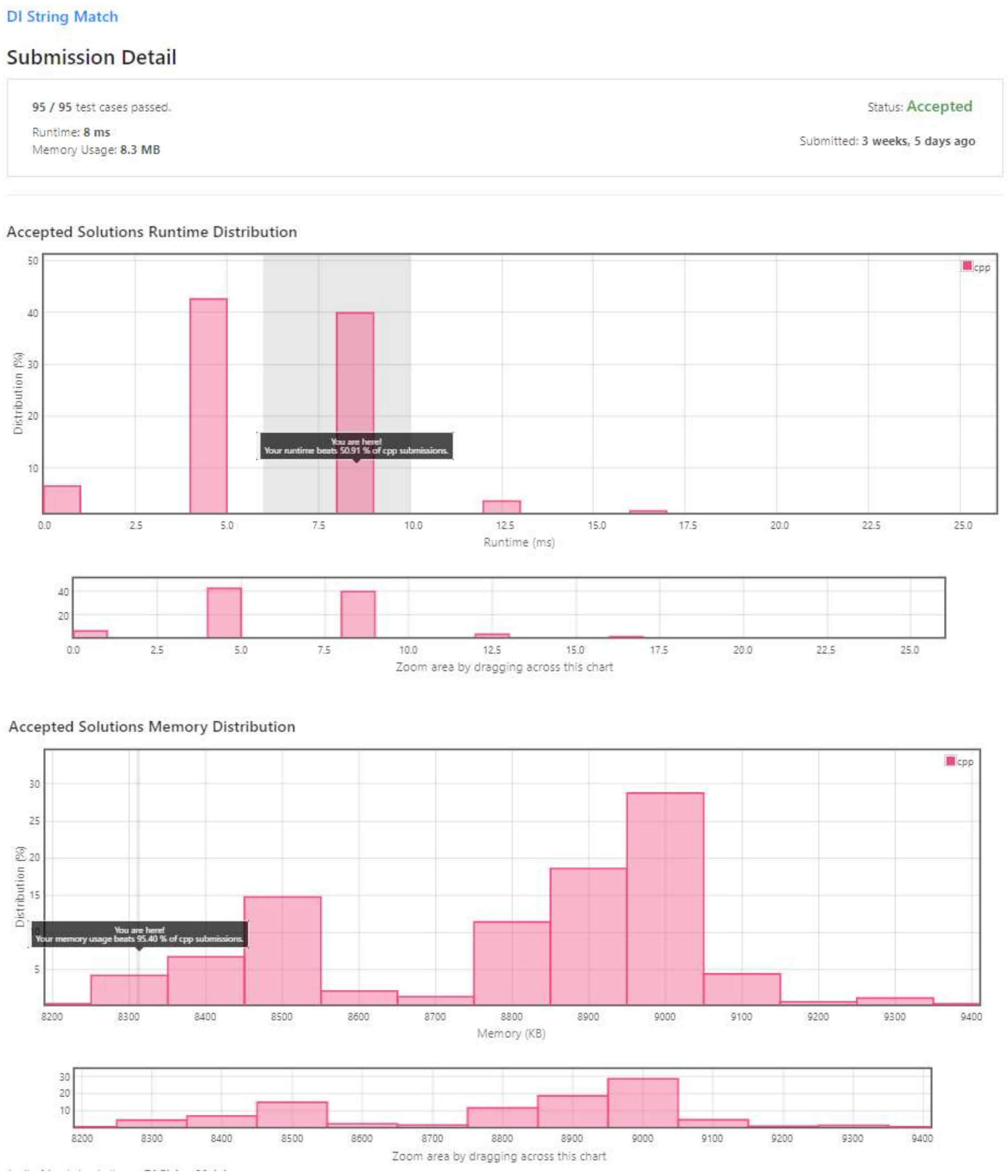
Example 3:

```
Input: "DDI"
Output: [3,2,0,1]
```

Note:

1. `1 <= S.length <= 10000`
2. `S` only contains characters `"I"` or `"D"`.

Solution 2:



DI String Match

Submission Detail

95 / 95 test cases passed.

Status: **Accepted**

Runtime: **8 ms**

Memory Usage: **8.3 MB**

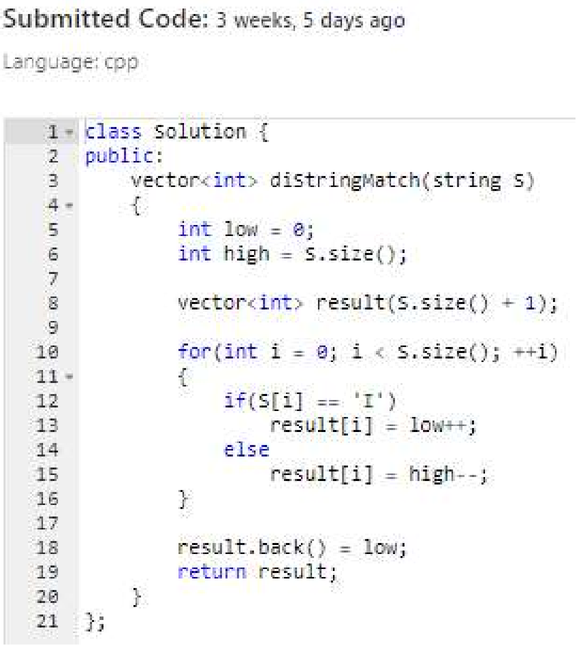
Submitted: **3 weeks, 5 days ago**

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 50.91 % of cpp submissions.

Accepted Solutions Memory Distribution

You are here!
Your memory usage beats 95.40 % of cpp submissions.

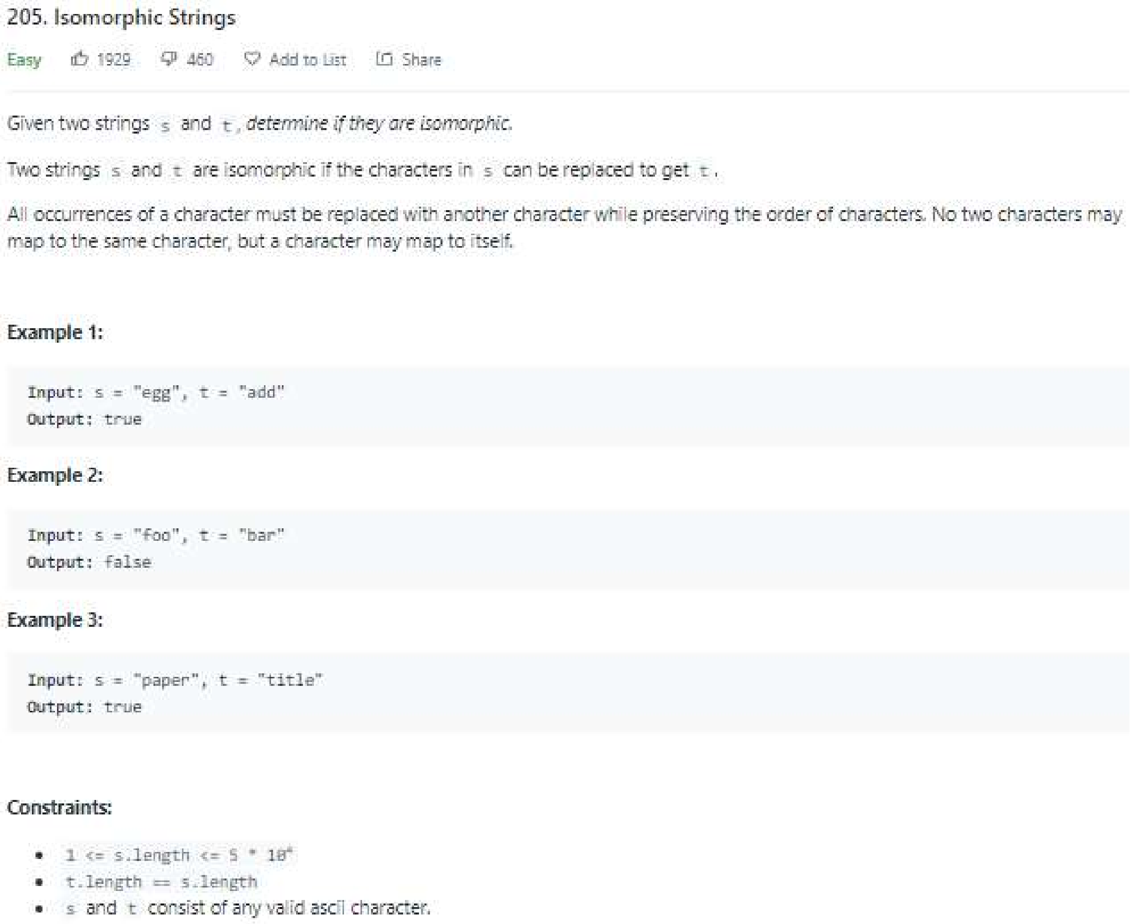


Submitted Code: 3 weeks, 5 days ago

Language: cpp

```cpp
class Solution {
public:
    vector<int> diStringMatch(string S)
    {
        int low = 0;
        int high = S.size();

        vector<int> result(S.size() + 1);

        for(int i = 0; i < S.size(); ++i)
        {
            if(S[i] == 'I')
                result[i] = low++;
            else
                result[i] = high--;
        }

        result.back() = low;
        return result;
    }
};
```

<u>Question 3:</u>



205. Isomorphic Strings

Given two strings `s` and `t`, *determine if they are isomorphic.*

Two strings `s` and `t` are isomorphic if the characters in `s` can be replaced to get `t`.

All occurrences of a character must be replaced with another character while preserving the order of characters. No two characters may map to the same character, but a character may map to itself.

Example 1:

```
Input: s = "egg", t = "add"
Output: true
```

Example 2:

```
Input: s = "foo", t = "bar"
Output: false
```

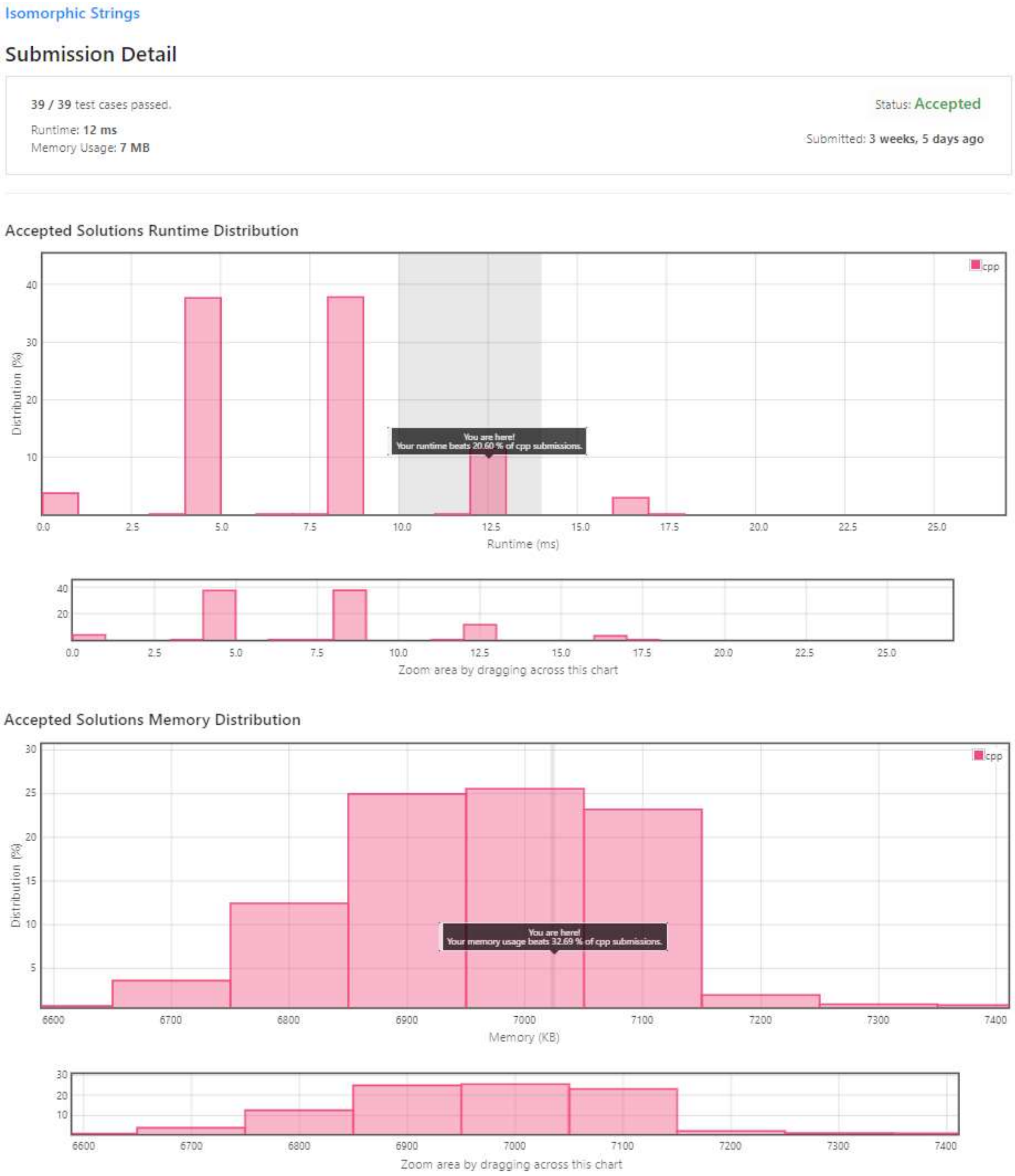
Example 3:

```
Input: s = "paper", t = "title"
Output: true
```

Constraints:

- `1 <= s.length <= 5 *` 10^4
- `t.length == s.length`
- `s` and `t` consist of any valid ascii character.

Solution 3:



Isomorphic Strings

Submission Detail

39 / 39 test cases passed.

Runtime: **12 ms**

Memory Usage: **7 MB**

Status: **Accepted**

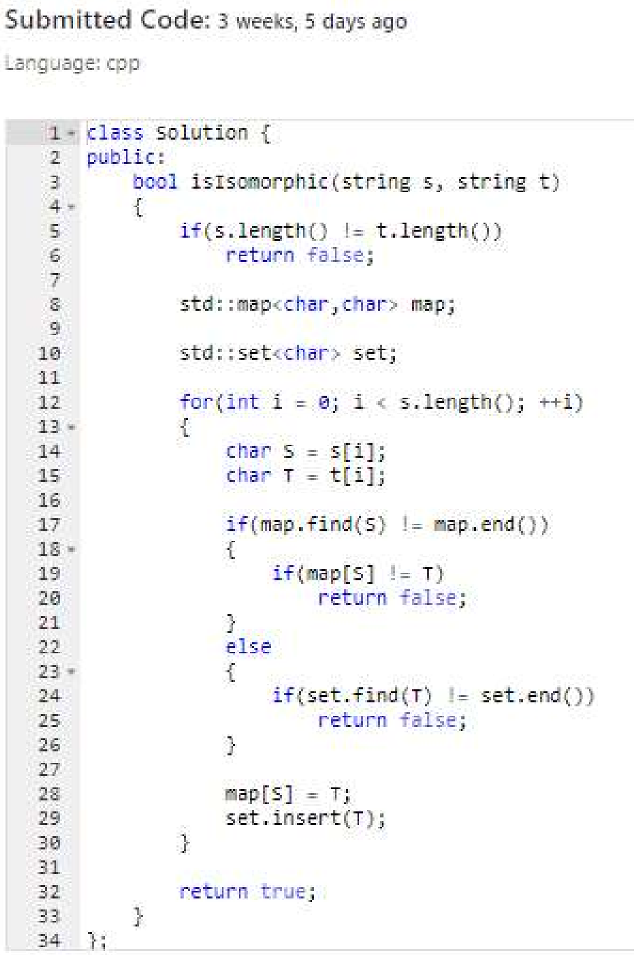
Submitted: **3 weeks, 5 days ago**

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 20.60 % of cpp submissions.

Accepted Solutions Memory Distribution

You are here!
Your memory usage beats 32.69 % of cpp submissions.

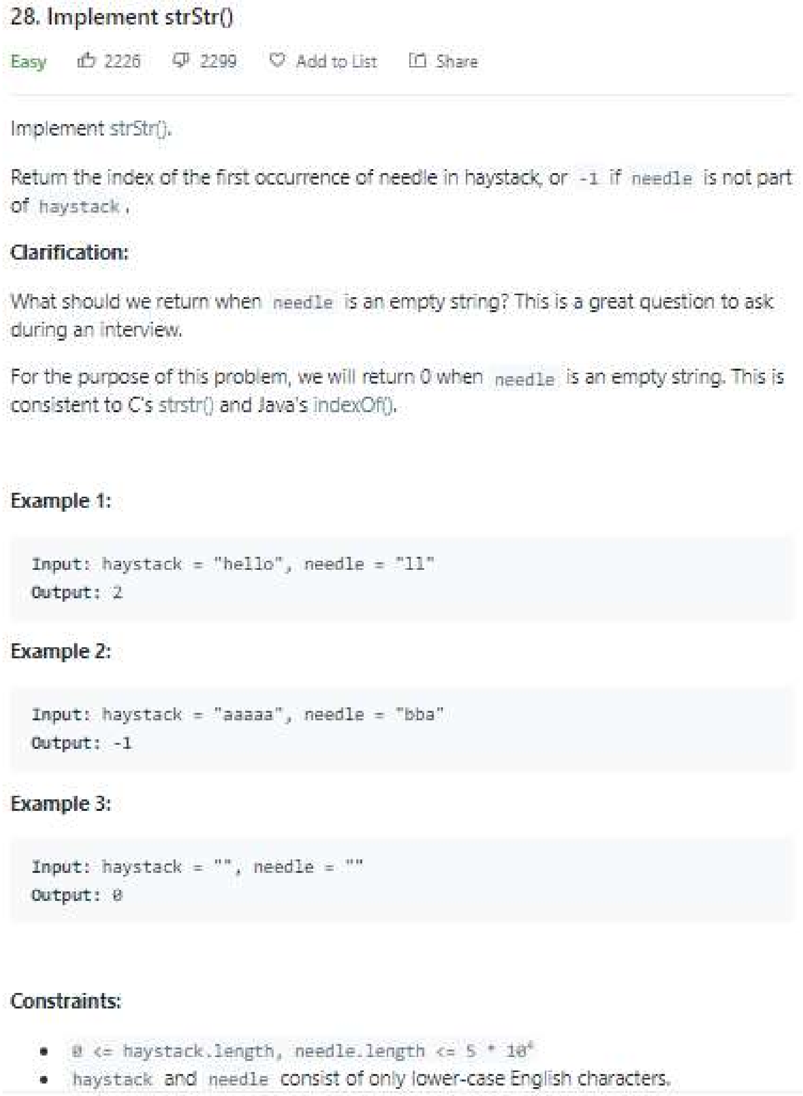


Submitted Code: 3 weeks, 5 days ago

Language: cpp

```cpp
class Solution {
public:
    bool isIsomorphic(string s, string t)
    {
        if(s.length() != t.length())
            return false;

        std::map<char,char> map;

        std::set<char> set;

        for(int i = 0; i < s.length(); ++i)
        {
            char S = s[i];
            char T = t[i];

            if(map.find(S) != map.end())
            {
                if(map[S] != T)
                    return false;
            }
            else
            {
                if(set.find(T) != set.end())
                    return false;
            }

            map[S] = T;
            set.insert(T);
        }

        return true;
    }
};
```

Question 4:



28. Implement strStr()

Easy 2226 2299 Add to List Share

Implement strStr().

Return the index of the first occurrence of needle in haystack, or `-1` if `needle` is not part of `haystack`.

Clarification:

What should we return when `needle` is an empty string? This is a great question to ask during an interview.

For the purpose of this problem, we will return 0 when `needle` is an empty string. This is consistent to C's strstr() and Java's indexOf().

Example 1:

```
Input: haystack = "hello", needle = "ll"
Output: 2
```

Example 2:

```
Input: haystack = "aaaaa", needle = "bba"
Output: -1
```

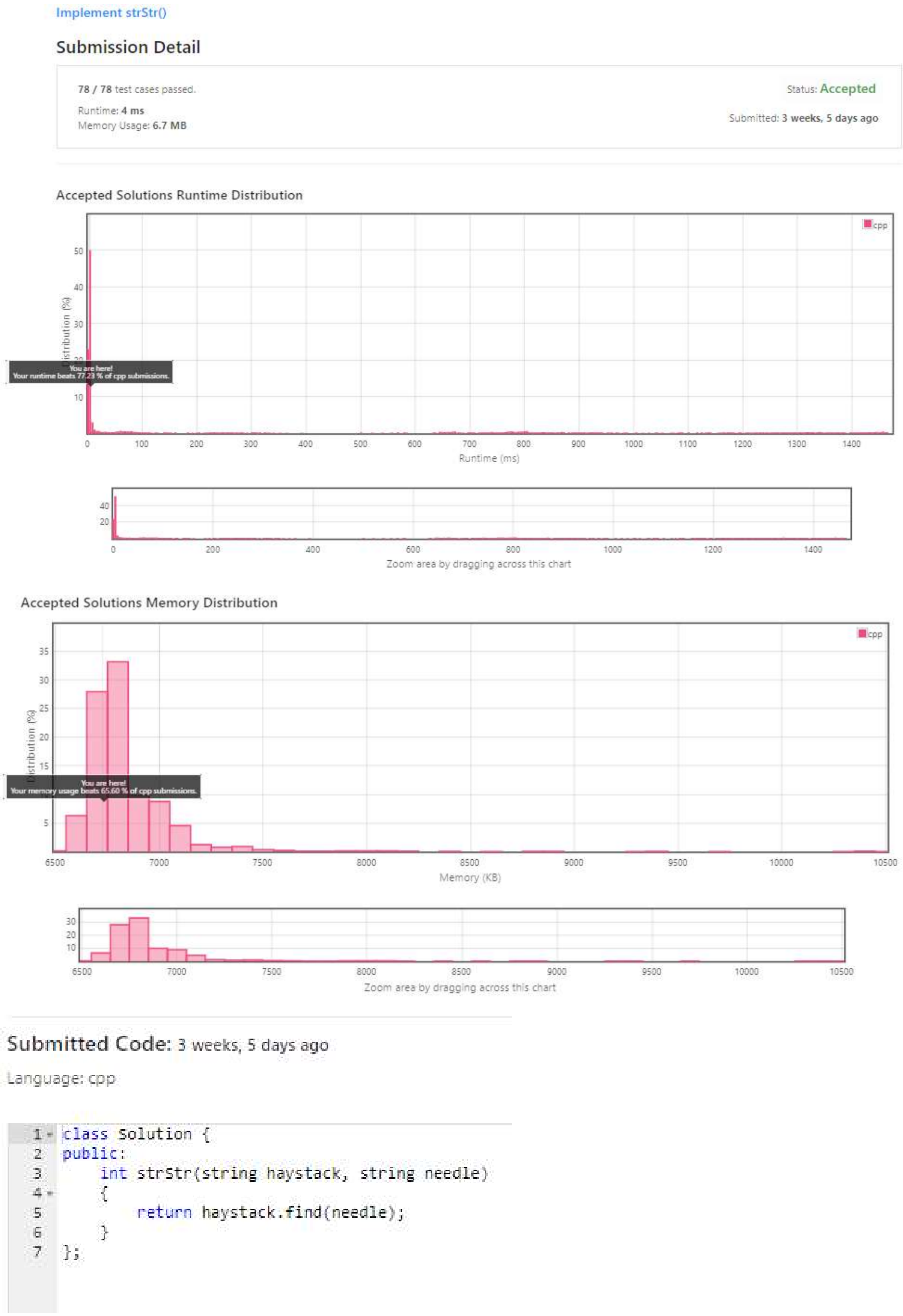
Example 3:

```
Input: haystack = "", needle = ""
Output: 0
```

Constraints:

- `0 <= haystack.length, needle.length <= 5 *` 10^4
- `haystack` and `needle` consist of only lower-case English characters.

Solution 4:



Implement strStr()

Submission Detail

78 / 78 test cases passed.

Status: Accepted

Runtime: 4 ms

Memory Usage: 6.7 MB

Submitted: 3 weeks, 5 days ago

Accepted Solutions Runtime Distribution

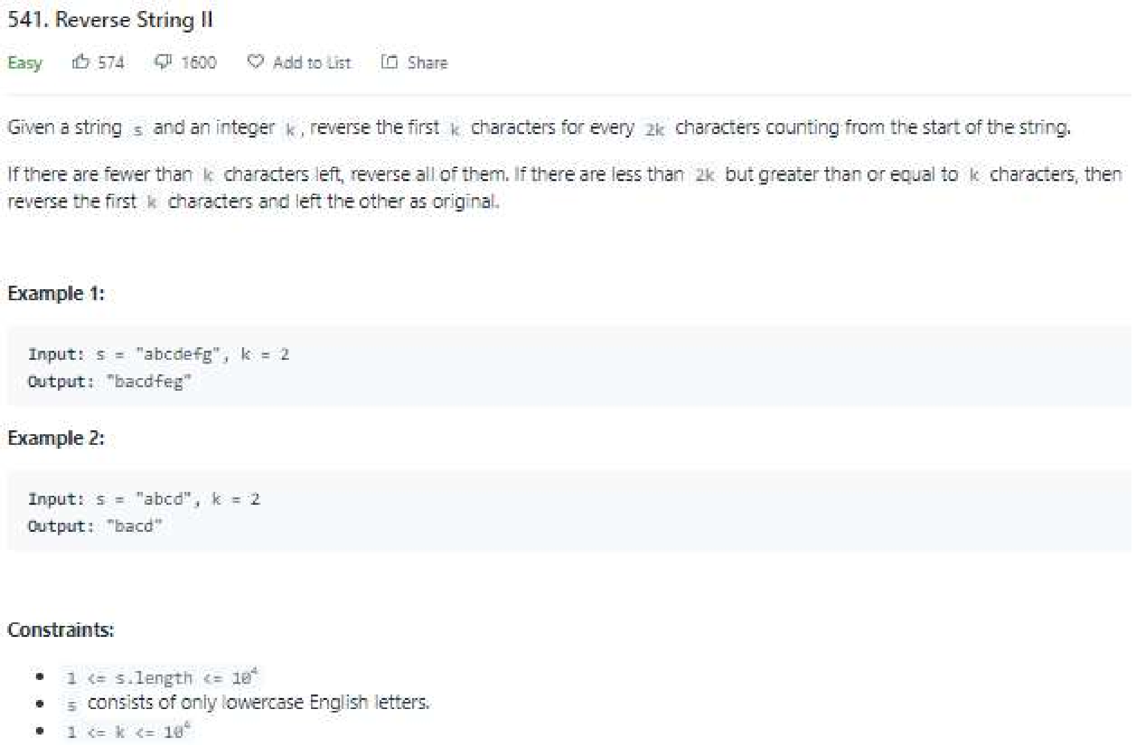
Accepted Solutions Memory Distribution

Submitted Code: 3 weeks, 5 days ago

Language: cpp

```cpp
class Solution {
public:
    int strStr(string haystack, string needle)
    {
        return haystack.find(needle);
    }
};
```

Question 5:



541. Reverse String II

Given a string `s` and an integer `k`, reverse the first `k` characters for every `2k` characters counting from the start of the string.

If there are fewer than `k` characters left, reverse all of them. If there are less than `2k` but greater than or equal to `k` characters, then reverse the first `k` characters and left the other as original.

Example 1:

```
Input: s = "abcdefg", k = 2
Output: "bacdfeg"
```

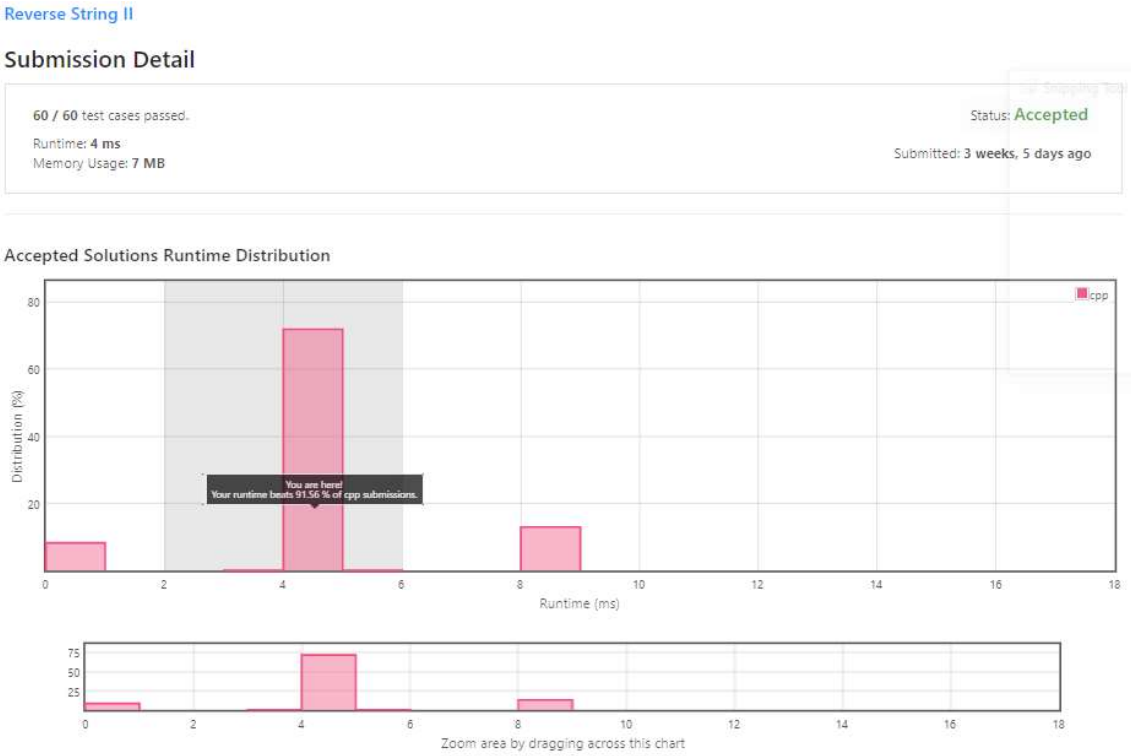
Example 2:

```
Input: s = "abcd", k = 2
Output: "bacd"
```

Constraints:

- $1 <= s.length <= 10^4$
- `s` consists of only lowercase English letters.
- $1 <= k <= 10^4$

Solution 5:



Reverse String II

Submission Detail

60 / 60 test cases passed.

Runtime: 4 ms

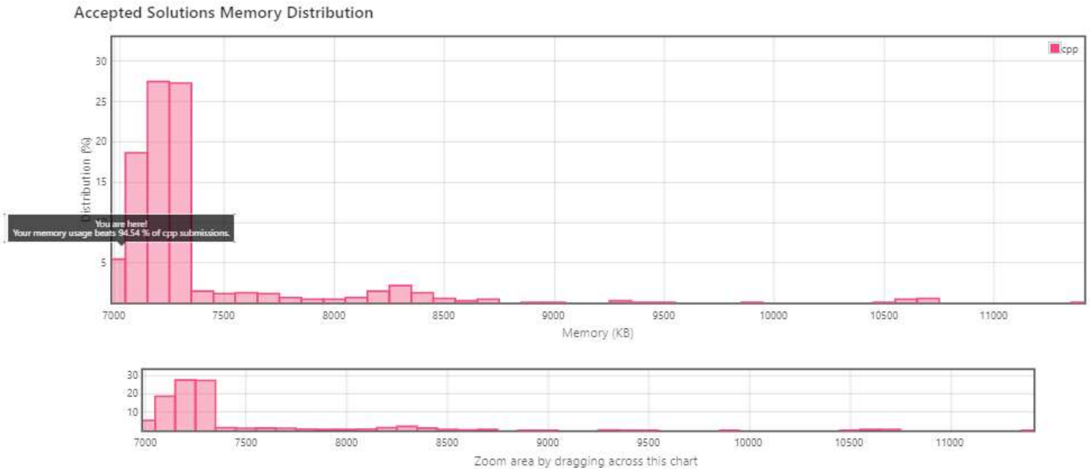
Memory Usage: 7 MB

Status: Accepted

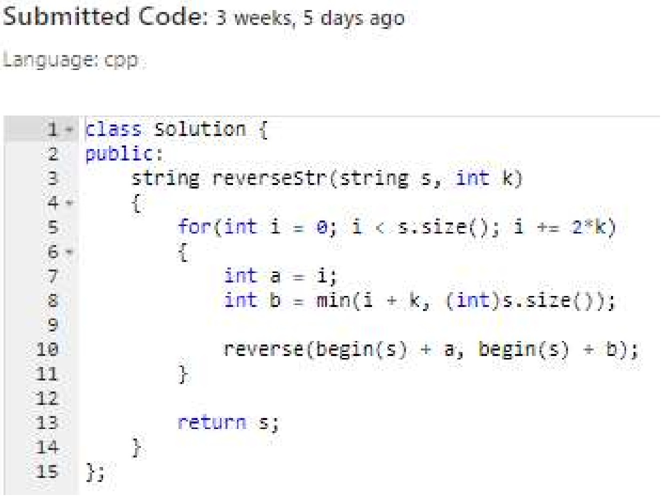
Submitted: 3 weeks, 5 days ago

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 91.56 % of cpp submissions.



You are here!
Your memory usage beats 94.54 % of cpp submissions.

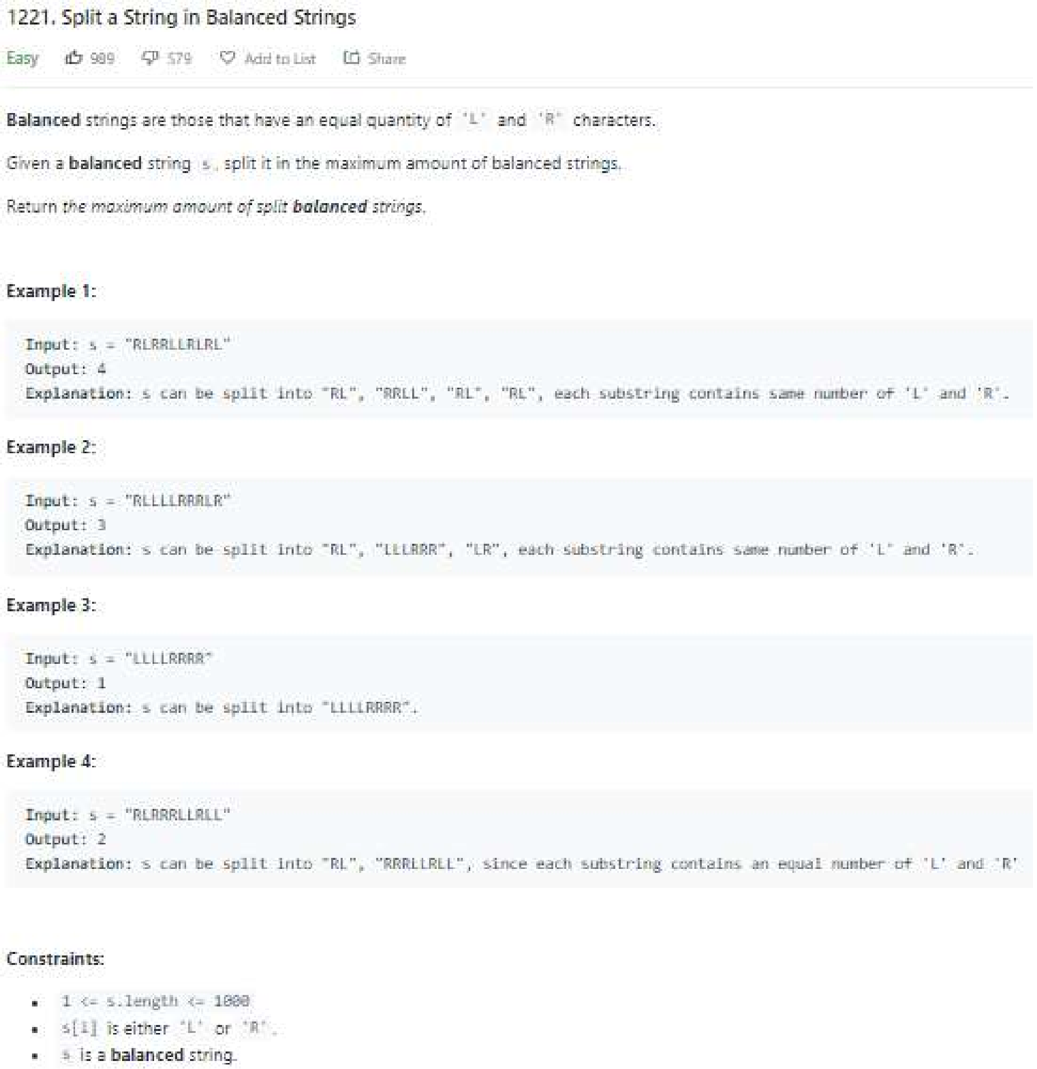


Submitted Code: 3 weeks, 5 days ago

Language: cpp

```cpp
class Solution {
public:
    string reverseStr(string s, int k)
    {
        for(int i = 0; i < s.size(); i += 2*k)
        {
            int a = i;
            int b = min(i + k, (int)s.size());

            reverse(begin(s) + a, begin(s) + b);
        }

        return s;
    }
};
```

Question 6:



1221. Split a String in Balanced Strings

Balanced strings are those that have an equal quantity of `'L'` and `'R'` characters.

Given a **balanced** string `s`, split it in the maximum amount of balanced strings.

Return *the maximum amount of split **balanced** strings.*

Example 1:

```
Input: s = "RLRRLLRLRL"
Output: 4
Explanation: s can be split into "RL", "RRLL", "RL", "RL", each substring contains same number of 'L' and 'R'.
```

Example 2:

```
Input: s = "RLLLLRRRLR"
Output: 3
Explanation: s can be split into "RL", "LLLRRR", "LR", each substring contains same number of 'L' and 'R'.
```

Example 3:

```
Input: s = "LLLLRRRR"
Output: 1
Explanation: s can be split into "LLLLRRRR".
```

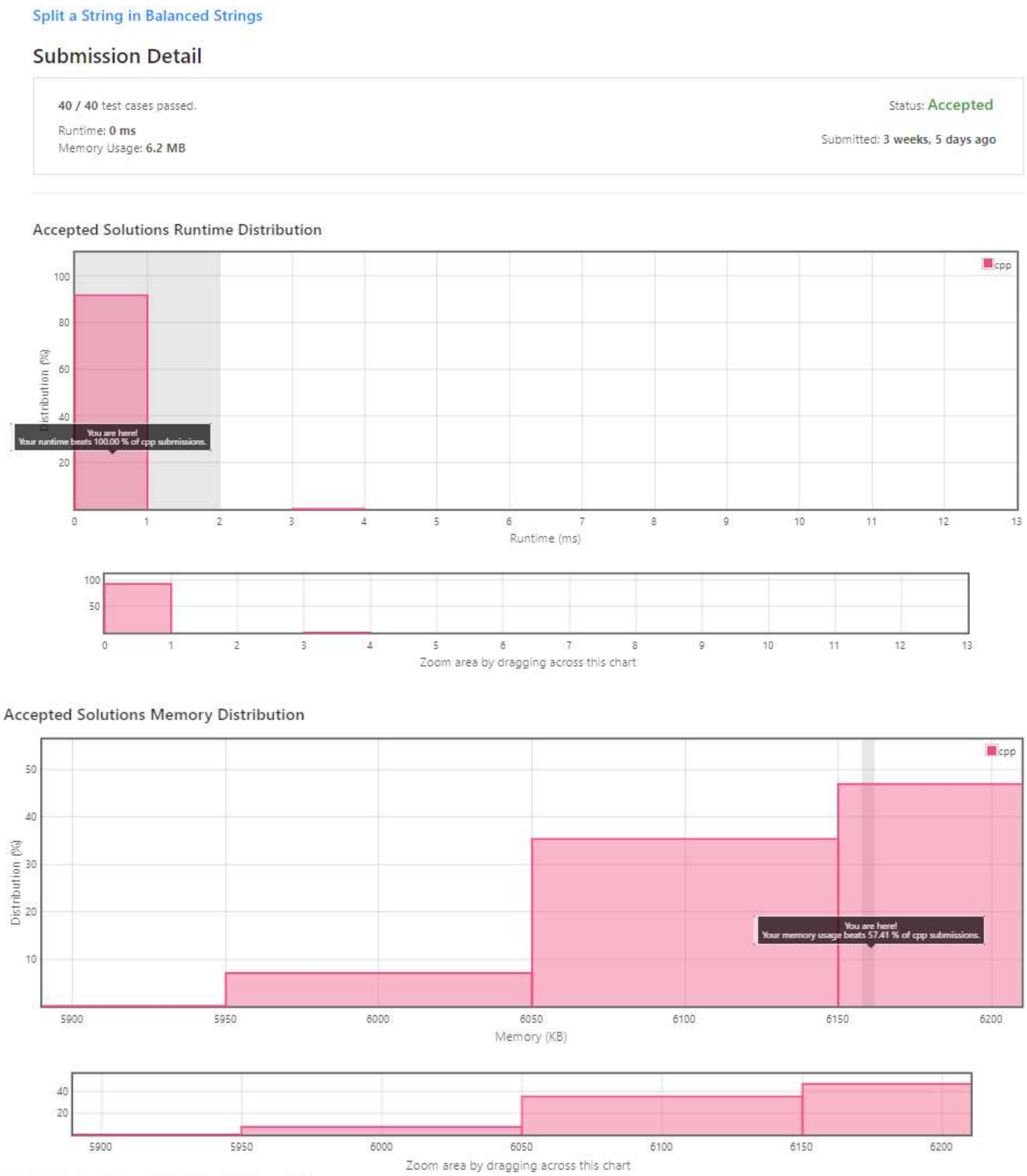
Example 4:

```
Input: s = "RLRRRLLRLL"
Output: 2
Explanation: s can be split into "RL", "RRRLLRLL", since each substring contains an equal number of 'L' and 'R'
```

Constraints:

- `1 <= s.length <= 1000`
- `s[i]` is either `'L'` or `'R'`.
- `s` is a **balanced** string.

Solution 6:



Split a String in Balanced Strings

Submission Detail

40 / 40 test cases passed.

Runtime: **0 ms**

Memory Usage: **6.2 MB**

Status: **Accepted**

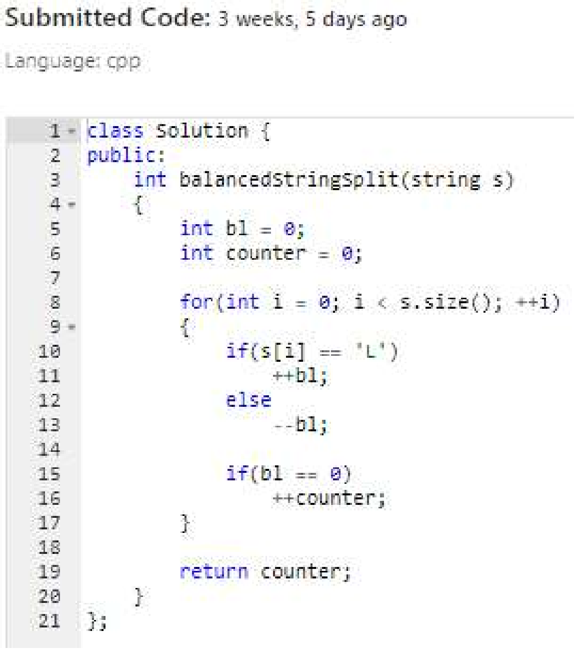
Submitted: **3 weeks, 5 days ago**

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 100.00 % of cpp submissions.

Accepted Solutions Memory Distribution

You are here!
Your memory usage beats 57.41 % of cpp submissions.

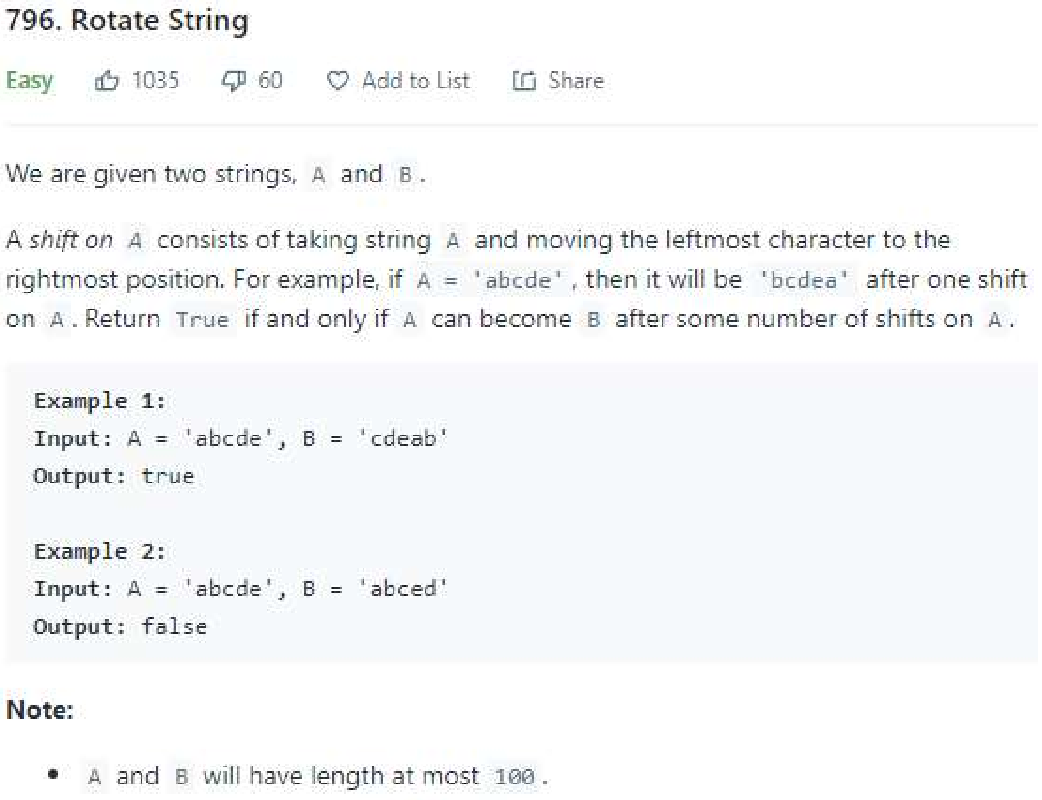


Submitted Code: 3 weeks, 5 days ago

Language: cpp

```cpp
class Solution {
public:
    int balancedStringSplit(string s)
    {
        int bl = 0;
        int counter = 0;
        
        for(int i = 0; i < s.size(); ++i)
        {
            if(s[i] == 'L')
                ++bl;
            else
                --bl;
            
            if(bl == 0)
                ++counter;
        }
        
        return counter;
    }
};
```

Question 7:



796. Rotate String

We are given two strings, `A` and `B`.

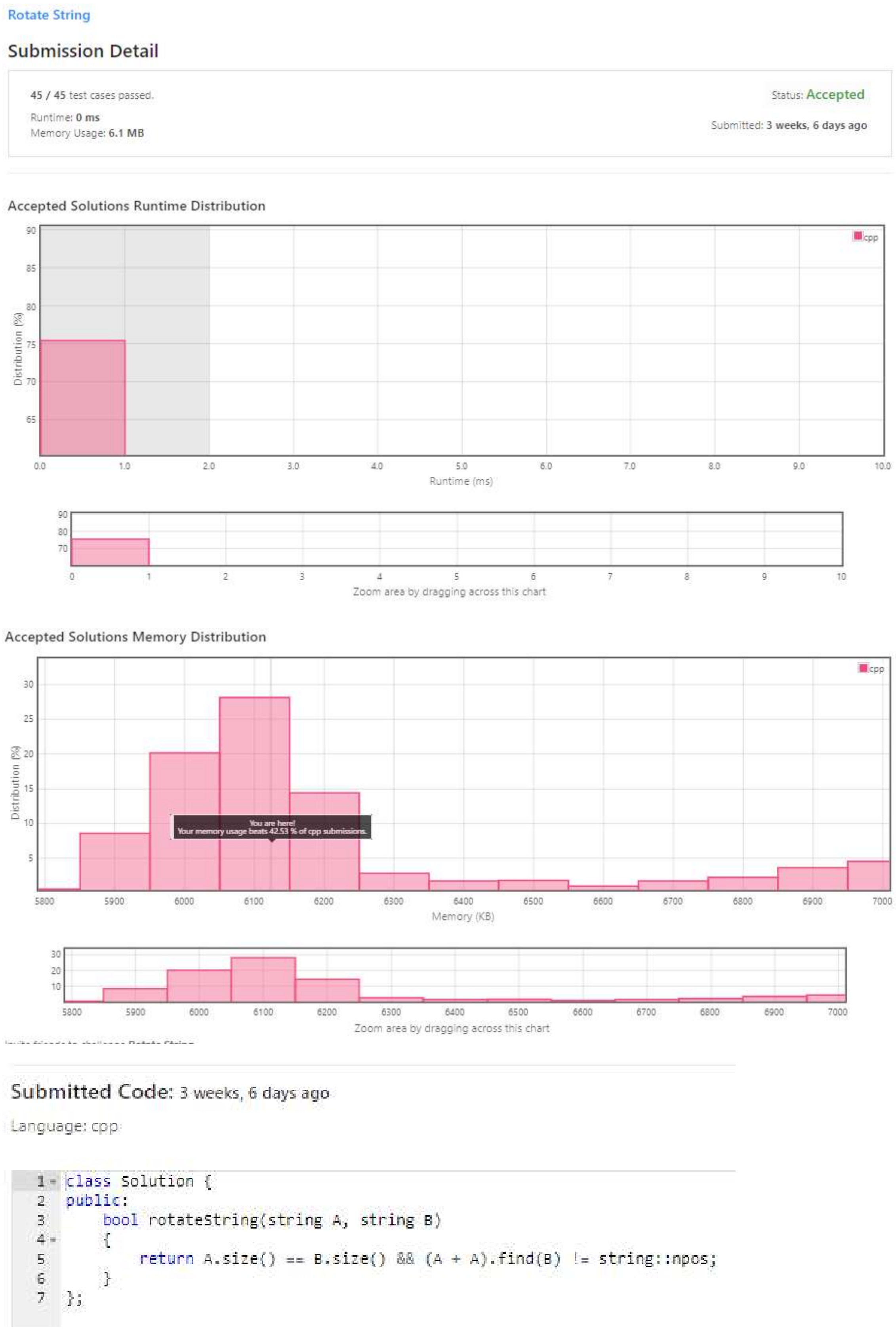
A *shift on* `A` consists of taking string `A` and moving the leftmost character to the rightmost position. For example, if `A = 'abcde'`, then it will be `'bcdea'` after one shift on `A`. Return `True` if and only if `A` can become `B` after some number of shifts on `A`.

```
Example 1:
Input: A = 'abcde', B = 'cdeab'
Output: true

Example 2:
Input: A = 'abcde', B = 'abced'
Output: false
```

Note:

- `A` and `B` will have length at most `100`.

Solution 7:



Rotate String

Submission Detail

45 / 45 test cases passed.

Runtime: **0 ms**

Memory Usage: **6.1 MB**

Status: Accepted

Submitted: **3 weeks, 6 days ago**

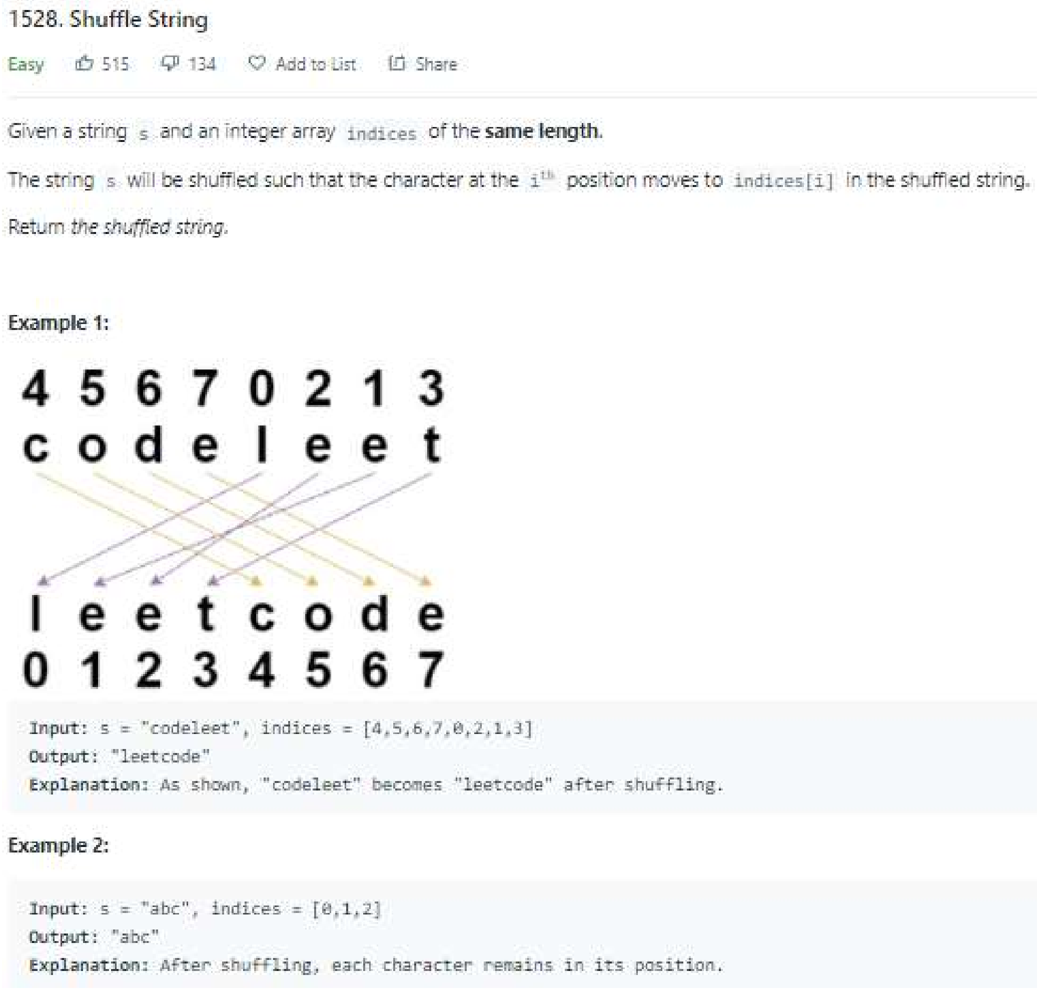
Invite friends to challenge **Rotate String**

Submitted Code: 3 weeks, 6 days ago

Language: cpp

```cpp
class Solution {
public:
    bool rotateString(string A, string B)
    {
        return A.size() == B.size() && (A + A).find(B) != string::npos;
    }
};
```

Question 8:



1528. Shuffle String

Given a string `s` and an integer array `indices` of the **same length**.

The string `s` will be shuffled such that the character at the `i`th position moves to `indices[i]` in the shuffled string.

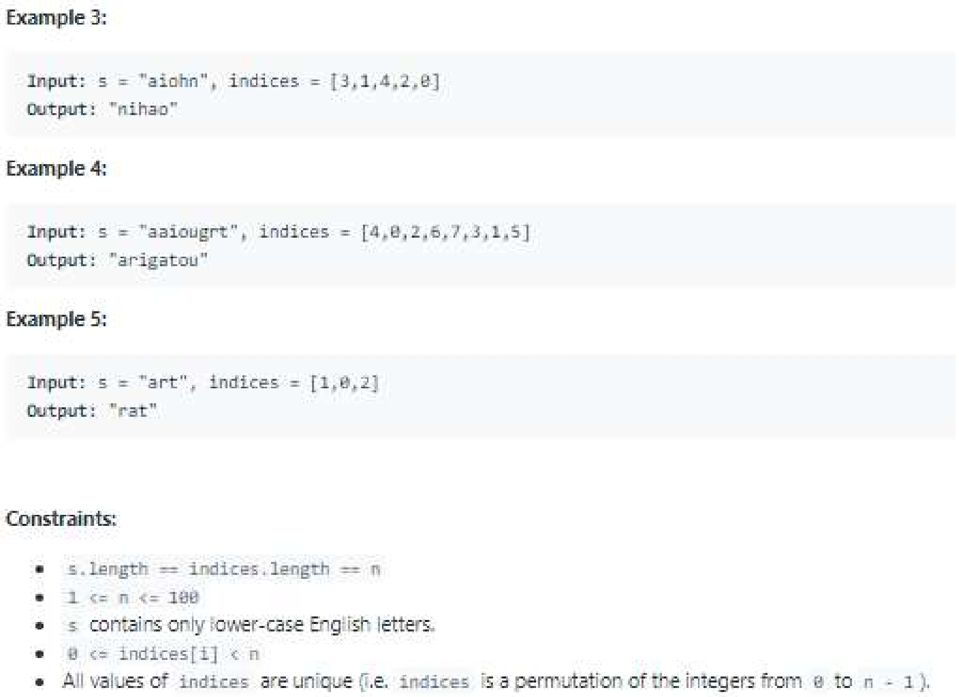
Return *the shuffled string*.

Example 1:

```
Input: s = "codeleet", indices = [4,5,6,7,0,2,1,3]
Output: "leetcode"
Explanation: As shown, "codeleet" becomes "leetcode" after shuffling.
```

Example 2:

```
Input: s = "abc", indices = [0,1,2]
Output: "abc"
Explanation: After shuffling, each character remains in its position.
```



Example 3:

```
Input: s = "aiohn", indices = [3,1,4,2,0]
Output: "nihao"
```

Example 4:

```
Input: s = "aaiougrt", indices = [4,0,2,6,7,3,1,5]
Output: "arigatou"
```

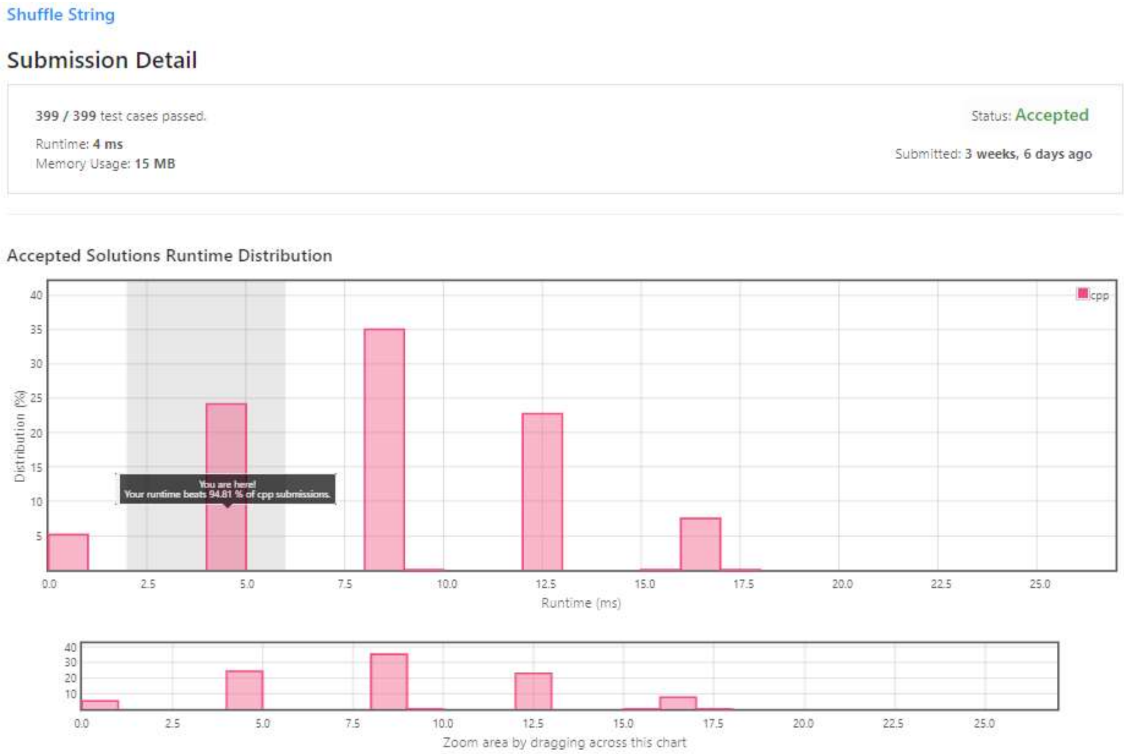
Example 5:

```
Input: s = "art", indices = [1,0,2]
Output: "rat"
```

Constraints:

- `s.length == indices.length == n`
- `1 <= n <= 100`
- `s` contains only lower-case English letters.
- `0 <= indices[i] < n`
- All values of `indices` are unique (i.e. `indices` is a permutation of the integers from `0` to `n - 1`).

Solution 8:



Shuffle String

Submission Detail

399 / 399 test cases passed.

Status: Accepted

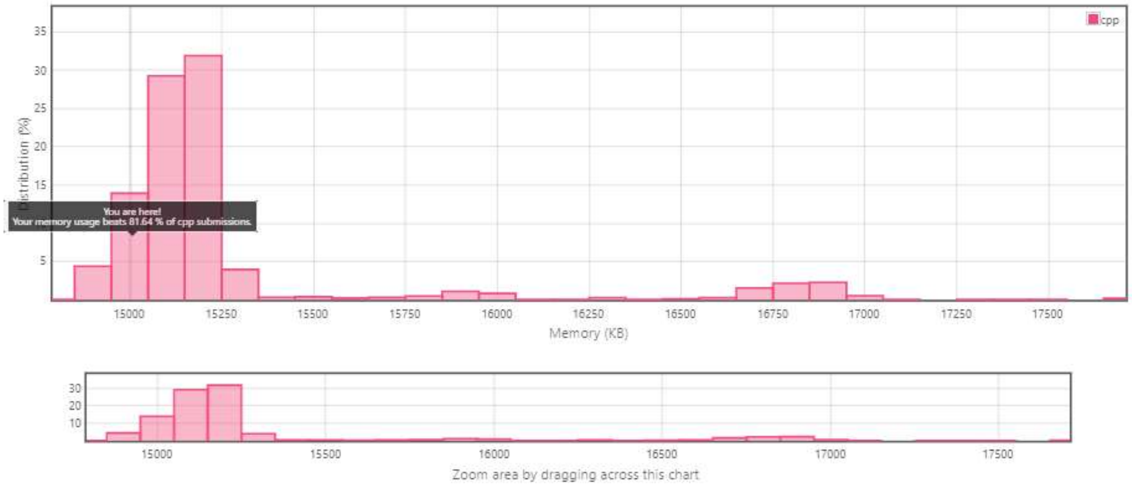
Runtime: 4 ms

Memory Usage: 15 MB

Submitted: 3 weeks, 6 days ago

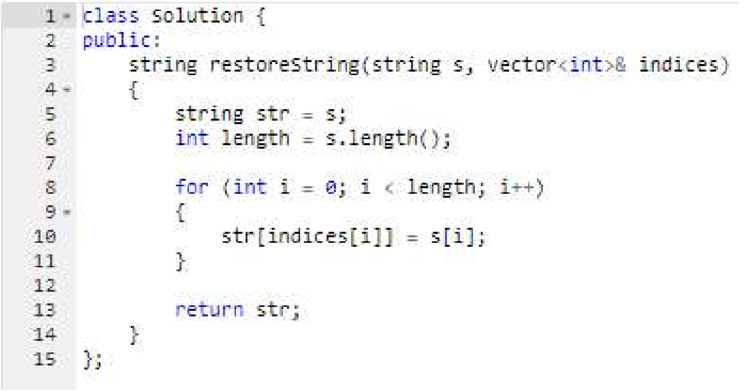
Accepted Solutions Runtime Distribution

Accepted Solutions Memory Distribution



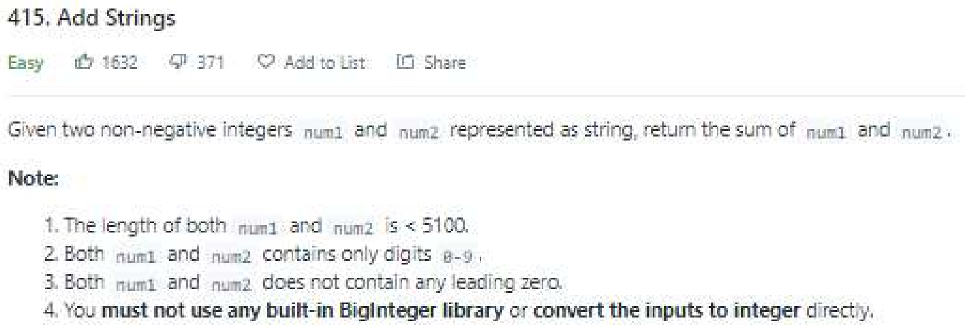
Submitted Code: 3 weeks, 6 days ago

Language: cpp



```cpp
class Solution {
public:
    string restoreString(string s, vector<int>& indices)
    {
        string str = s;
        int length = s.length();

        for (int i = 0; i < length; i++)
        {
            str[indices[i]] = s[i];
        }

        return str;
    }
};
```

Question 9:



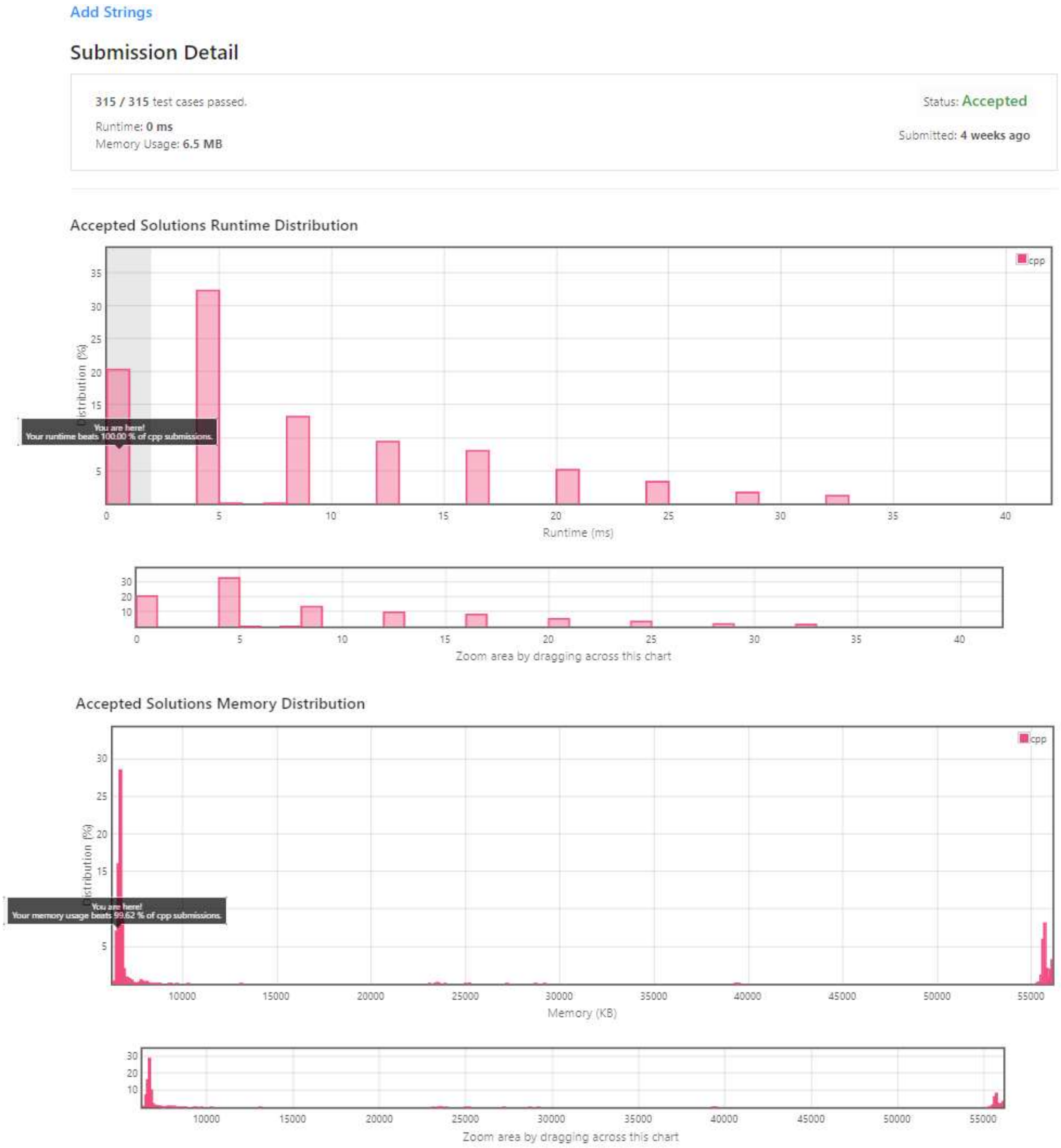
415. Add Strings

Given two non-negative integers `num1` and `num2` represented as string, return the sum of `num1` and `num2`.

Note:

1. The length of both `num1` and `num2` is < 5100.
2. Both `num1` and `num2` contains only digits `0-9`.
3. Both `num1` and `num2` does not contain any leading zero.
4. You **must not use any built-in BigInteger library** or **convert the inputs to integer** directly.

Solution 9:



Add Strings

Submission Detail

315 / 315 test cases passed.

Runtime: 0 ms

Memory Usage: 6.5 MB

Status: Accepted

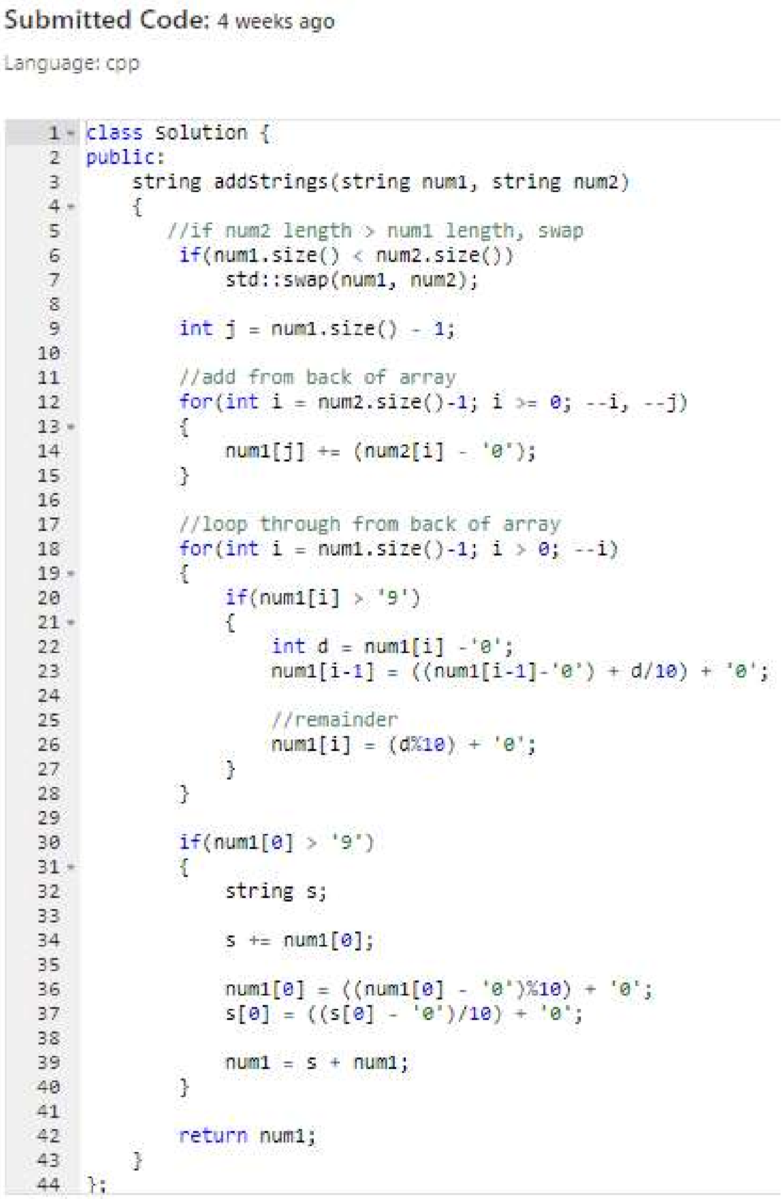
Submitted: 4 weeks ago

Accepted Solutions Runtime Distribution

You are here!
Your runtime beats 100.00 % of cpp submissions.

Accepted Solutions Memory Distribution

You are here!
Your memory usage beats 99.62 % of cpp submissions.

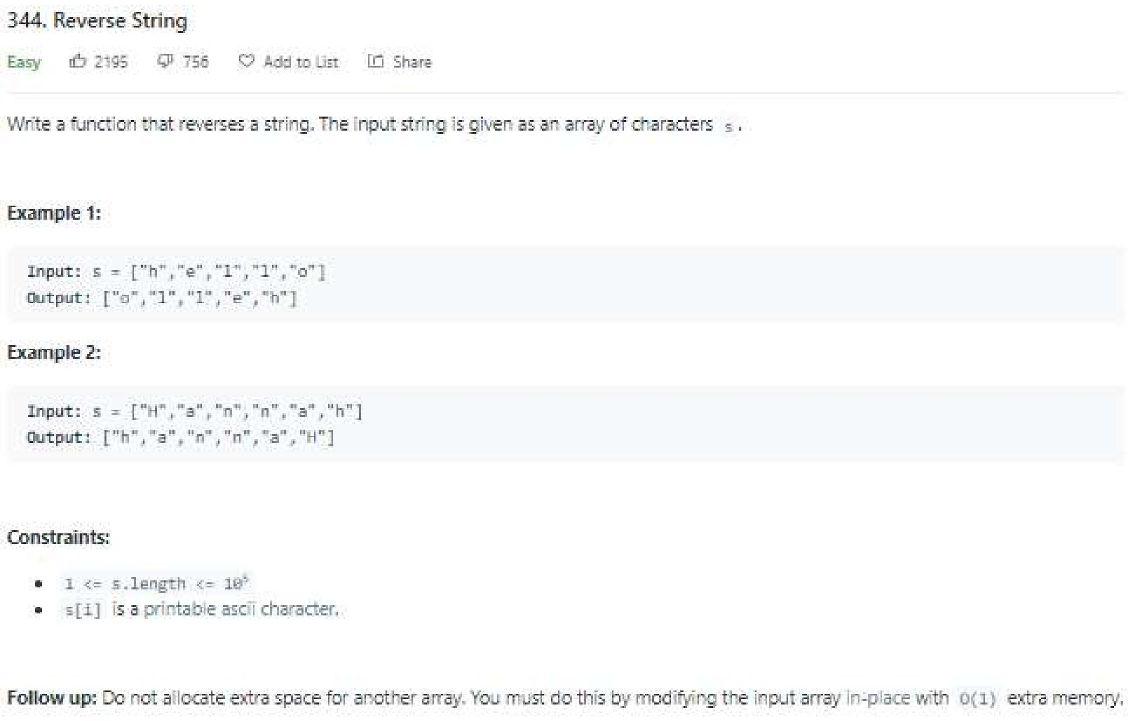


Submitted Code: 4 weeks ago

Language: cpp

```cpp
class Solution {
public:
    string addStrings(string num1, string num2)
    {
       //if num2 length > num1 length, swap
        if(num1.size() < num2.size())
            std::swap(num1, num2);

        int j = num1.size() - 1;

        //add from back of array
        for(int i = num2.size()-1; i >= 0; --i, --j)
        {
            num1[j] += (num2[i] - '0');
        }

        //loop through from back of array
        for(int i = num1.size()-1; i > 0; --i)
        {
            if(num1[i] > '9')
            {
                int d = num1[i] -'0';
                num1[i-1] = ((num1[i-1]-'0') + d/10) + '0';

                //remainder
                num1[i] = (d%10) + '0';
            }
        }

        if(num1[0] > '9')
        {
            string s;

            s += num1[0];

            num1[0] = ((num1[0] - '0')%10) + '0';
            s[0] = ((s[0] - '0')/10) + '0';

            num1 = s + num1;
        }

        return num1;
    }
};
```

Question 10:



344. Reverse String

Write a function that reverses a string. The input string is given as an array of characters `s`.

Example 1:

```
Input: s = ["h","e","l","l","o"]
Output: ["o","l","l","e","h"]
```

Example 2:

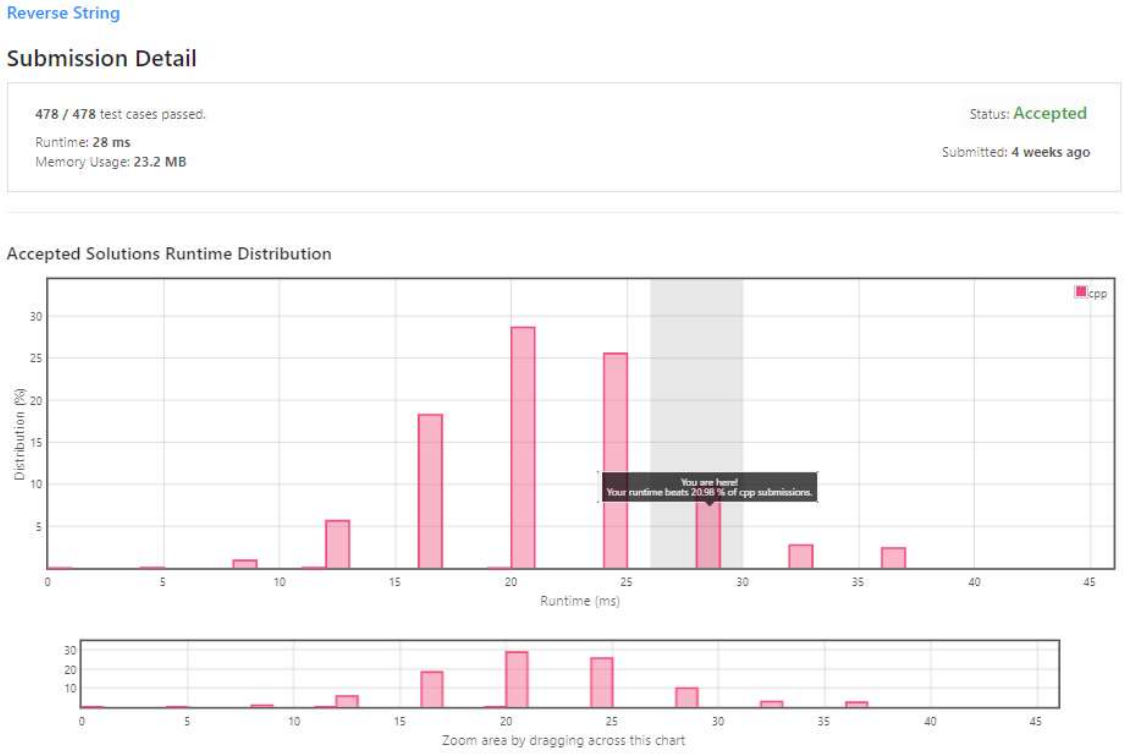
```
Input: s = ["H","a","n","n","a","h"]
Output: ["h","a","n","n","a","H"]
```

Constraints:

- `1 <= s.length <=` 10^5
- `s[i]` is a printable ascii character.

Follow up: Do not allocate extra space for another array. You must do this by modifying the input array in-place with `O(1)` extra memory.

Solution 10:



Reverse String

Submission Detail

478 / 478 test cases passed.

Runtime: 28 ms

Memory Usage: 23.2 MB

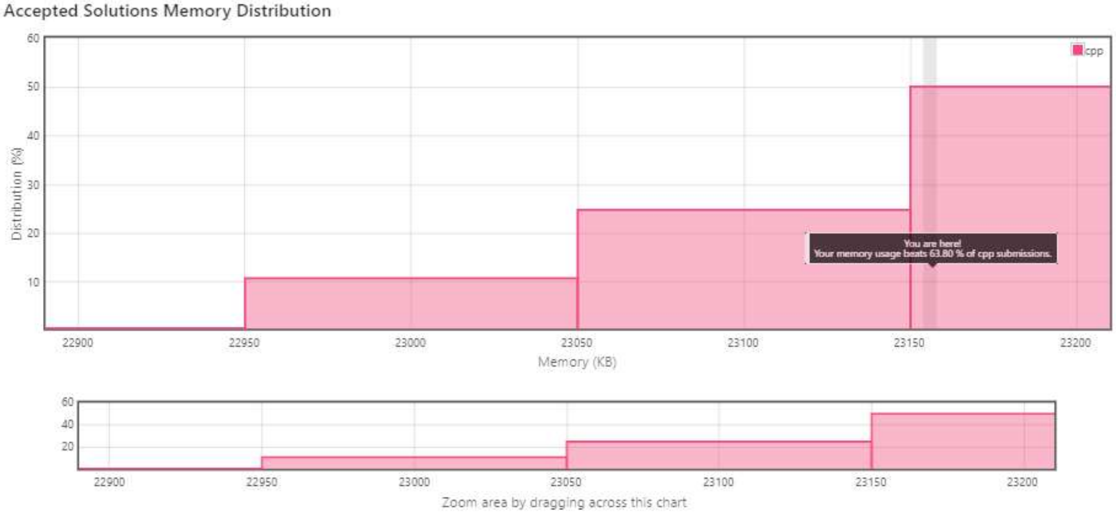
Status: Accepted

Submitted: 4 weeks ago

Accepted Solutions Runtime Distribution

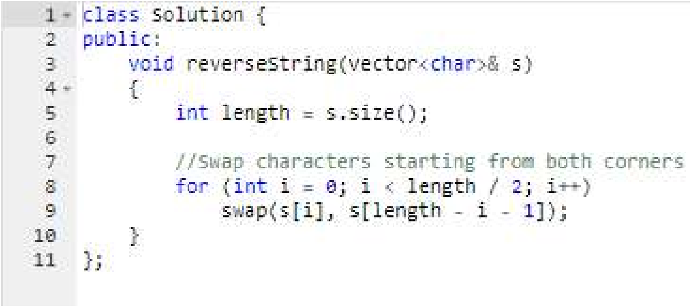
cpp
Distribution (%)
30
25
20
15
10
5
0
5
10
15
20
25
30
35
40
45
Runtime (ms)
You are here!
Your runtime beats 20.98 % of cpp submissions.
30
20
10
Zoom area by dragging across this chart

Accepted Solutions Memory Distribution



Submitted Code: 4 weeks ago

Language: cpp



```cpp
class Solution {
public:
    void reverseString(vector<char>& s)
    {
        int length = s.size();

        //Swap characters starting from both corners
        for (int i = 0; i < length / 2; i++)
            swap(s[i], s[length - i - 1]);
    }
};
```