

Fecha: 24-09-21

Actividad 2.2 - Screen Shots

Equipo:

- Jeshua Nava Avila | A01639282
- Barbara Nicole Vidal Sandoval | A01635233
- Renet de Jesús Pérez Gómez | A01640555

Valid parenthesis

The screenshot shows the LeetCode interface for the 'Valid Parentheses' problem. The submission is successful, with a runtime of 0 ms and memory usage of 6.2 MB. The C++ code is displayed in the editor, showing a linked list-based solution for validating parentheses.

Success Details >

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Valid Parentheses.

Memory Usage: 6.2 MB, less than 96.50% of C++ online submissions for Valid Parentheses.

Next challenges:

- Generate Parentheses
- Longest Valid Parentheses
- Remove Invalid Parentheses
- Check If Word Is Valid After Substitutions

Show off your acceptance: [f](#) [t](#) [in](#)

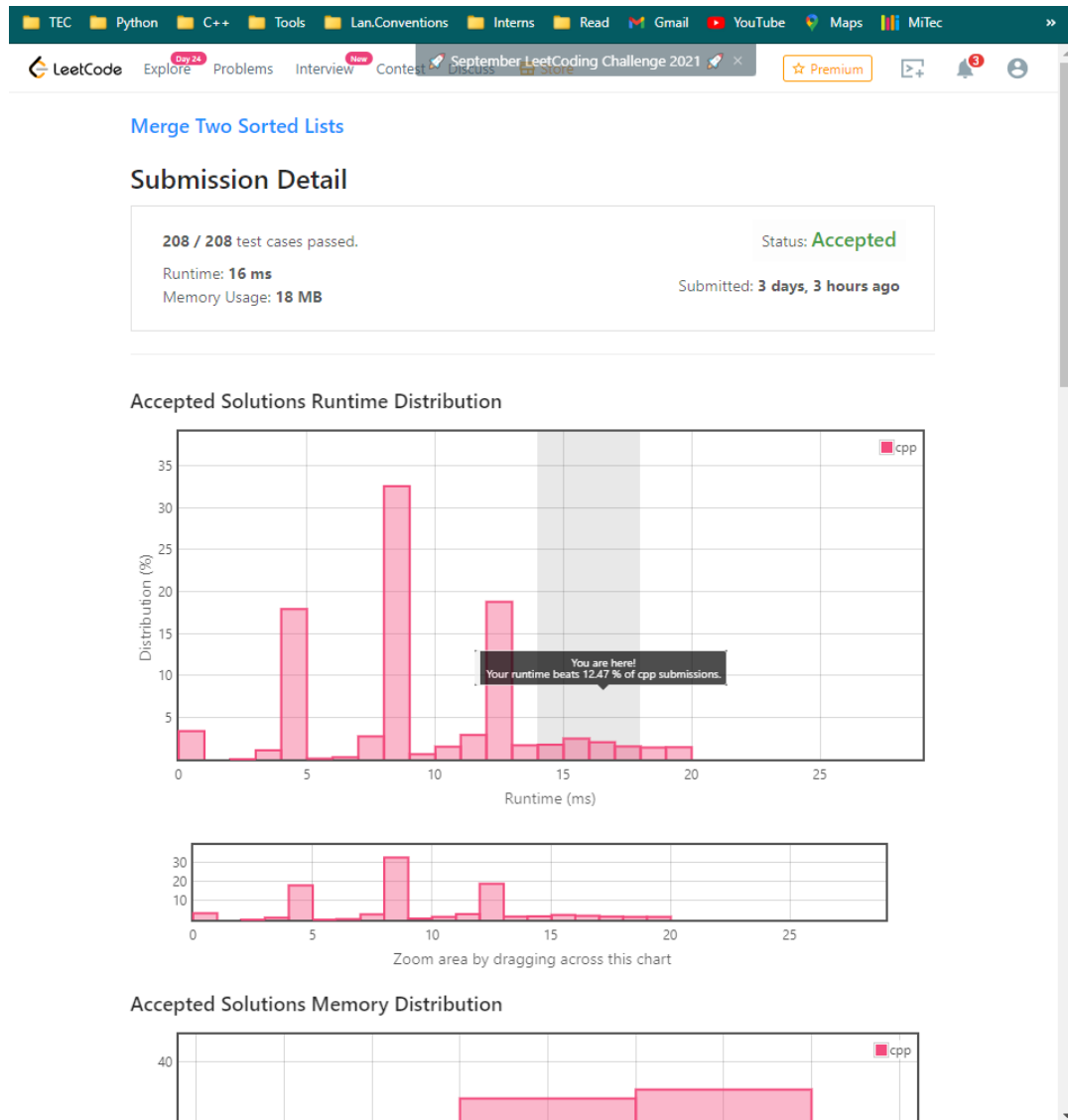
Time Submitted	Status	Runtime	Memory	Language
09/24/2021 16:16	Accepted	0 ms	6.2 MB	cpp
09/24/2021 16:16	Accepted	0 ms	6.2 MB	cpp

```
1  template <class T> class Node {
2      private:
3          T data; //Information of this node
4          Node* next; //Reference to next node in list
5
6      public:
7          //Constructor
8          Node(T new_data, Node *next_node){
9              this->data = new_data;
10             this->next = next_node;
11         }
12
13         //Default constructor
14         Node(T new_data){
15             this->data = new_data;
16             this->next = NULL;
17         }
18
19         void set_data(T new_data){
20             this->data = new_data;
21         }
22
23         T get_data(){
24             return this->data;
25         }
26
27         T get_afterNode_data(){
28             return this->next->get_data();
29         }
30     };
```

Your previous code was restored from your local storage. [Reset to default](#)

Console ▾ Contribute *i* [Run Code](#) [Submit](#)

Merge two sorted list



Palindrome Number

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

Success

Details

Runtime: 20 ms, faster than 27.11% of C++ online submissions for Palindrome Number.

Memory Usage: 12.1 MB, less than 6.23% of C++ online submissions for Palindrome Number.

Next challenges:

Palindrome Linked List

Show off your acceptance:

Time Submitted	Status	Runtime	Memory	Language
09/24/2021 15:34	Accepted	20 ms	12.1 MB	cpp
09/22/2021 14:33	Accepted	20 ms	12.1 MB	cpp

Problems

Pick One

Prev

9/2015

Next

C++

Autocomplete

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

asc_node = asc_node->get_next();

desc_node = desc_node->get_previous();

}

}

return true;

};

};

class Solution {

public:

bool isPalindrome(int x) {

std::string chain = std::to_string(x);

char verifyNegative = chain[0];

if(verifyNegative=='-')

return false;

CustomDoubleLinkedList<char> doubleLinkedList;

for (int i = 0; i < chain.length(); i++){

doubleLinkedList.add_end(chain[i]);

}

return doubleLinkedList.verifyPalindrome();

}

};

Your previous code was restored from your local storage. [Reset to default](#)

Console

Contribute

Run Code

Submit

Reverse Integer

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

7. Reverse Integer

Easy

5698

8511

Add to List

Share

Given a signed 32-bit integer x , return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range $[-2^{31}, 2^{31} - 1]$, then return 0.

Assume the environment does not allow you to store 64-bit integers (signed or unsigned).

Example 1:

Input: $x = 123$

Output: 321

Example 2:

Input: $x = -123$

Output: -321

Example 3:

Input: $x = 120$

C++

Autocomplete

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

#include <iostream>

#include <stream>

#include <string>

#include <vector>

#include <iterator>

#include <cmath>

using namespace std;

class Solution {

public:

int reverse(int x) {

int num=2;

if(x>(pow(num,31)-1) || x<pow(-num,31)){ //x no se tiene que pasar de esos rangos

cout<<"Numero invalido"<<endl;

return 0;

}

else{

string xs = to_string(x); //Convierte a string el valor de x

if(x>0){ //si el valor de x es mayor o igual a 0...en caso de ser un número

positivo

cout<<"Valor positivo: "<<x<<endl;

string xs_reversed(xs.rbegin(), xs.rend()); // se invierte el string x

string xsnew = xs_reversed; // se crea una nueva variable con el string

invertido

stringstream geek(xsnew); //Se convierte de string a int

int xi = 0;

geek >> xi;

if(xi>(pow(num,31)-1) || xi<pow(-num,31)){ //xi no se tiene que pasar de

esos rangos

cout<<"Resultado invalido"<<endl;

return 0;

}

else{

Console

Contribute

Run Code

Submit

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

7. Reverse Integer

Easy

5698

8511

Add to List

Share

Given a signed 32-bit integer x , return x with its digits reversed. If reversing x causes the value to go outside the signed 32-bit integer range $[-2^{31}, 2^{31} - 1]$, then return 0.

Assume the environment does not allow you to store 64-bit integers (signed or unsigned).

Example 1:

Input: $x = 123$

Output: 321

Example 2:

Input: $x = -123$

Output: -321

Example 3:

Input: $x = 120$

C++

Autocomplete

23

24

25

26

27

28

29

30

31

32

33

34

35

string xsnew = xs_reversed; // se crea una nueva variable con el string

invertido

stringstream geek(xsnew); //Se convierte de string a int

int xi = 0;

geek >> xi;

if(xi>(pow(num,31)-1) || xi<pow(-num,31)){ //xi no se tiene que pasar de

esos rangos

cout<<"Resultado invalido"<<endl;

return 0;

}

else{

cout<<"Resultado: ";

return xi; //regresa el valor invertido

}

Testcase

Run Code Result

Debugger

Accepted

Runtime: 4 ms

Your input

123

stdout

Valor positivo: 123

Resultado:

Output

321

Diff

Expected

321

Problems

Pick One

Prev

7/2015

Next

Console

Use Example Testcases

Run Code

Submit