

7 |Day4

30

7

“ ”

4

-
- LRU
- a z 26 Trie
-

LeetCode @Smallfly

- Reverse String
<https://leetcode.com/problems/reverse-string/>
<https://leetcode-cn.com/problems/reverse-string/>
- Reverse Words in a String
<https://leetcode.com/problems/reverse-words-in-a-string/>
<https://leetcode-cn.com/problems/reverse-words-in-a-string/>

- String to Integer (atoi) (atoi)

<https://leetcode.com/problems/string-to-integer-atoi/>

<https://leetcode-cn.com/problems/string-to-integer-atoi/>

“ ”

- 2019-02-08 07:34:10

murmur

lua table

+

8Um{

2

1.5

1.25.....

golang

rehash

redis

[2]

- kai 2019-02-08 21:15:31

~ [1]

- 2019-02-10 00:14:08

python

import math

class Solution:

def myAtoi(self, str: 'str') -> 'int':

result = 0

i, N, former = 0, len(str), 1

while i < N:

if str[i] != ' ':

break

i += 1

if i < N and (str[i] == '-' or str[i] == '+):

former = -1 if str[i] == '-' else 1

i += 1

while i < N:

if str[i].isdigit():

result = result * 10 + int(str[i])

i += 1

else:

break

```
result = result * former
if result > (math.pow(2, 31) * -1) and result < (math.pow(2,31) - 1):
    return result
elif former > 0 :
    return int(math.pow(2,31) - 1)
else:
    return int(math.pow(2,31) * -1)
```

• 2019-02-09 18:57:59

1. $O(n)$
2. $O(n^2)$
 $s \quad t \quad s \quad t \quad s \quad t$
 $s \quad t \quad O(n^2)$
3. 0 “_” $int \ num=0$ $a \quad a \quad num$
 $*=10 \quad num+=a-'0'$ “_” $num* \ -1 \quad num \quad int$
 $10 \quad “_”$

• 2019-02-09 16:19:19

```
python
class Solution:
def reverseString(self, s: 'List[str]') -> 'None':
    """
    Do not return anything, modify s in-place instead.
    """
    i, N = 0, len(s)
    while i < N//2:
        s[i], s[N-1-i] = s[N-1-i], s[i]
        i += 1

    print(s)
```

https://blog.csdn.net/github_38313296/article/details/86818634

- ext4 2019-02-09 07:37:12

```
class Solution {
public:
    string reverseString(string s) {
        int length = s.length();
        if (length < 2) {
            return s;
        }
        int i = 0, j = length - 1;
        char temp;
        while (i < j) {
            temp = s[i];
            s[i] = s[j];
            s[j] = temp;
            i++;
            j--;
        }
        return s;
    }
};
```

- 2019-02-08 23:18:33

java hashmap

hashmap

leetcode

1. Reverse String

O(1)

s[0...n-1]

i<n/2; i n-1-i

.

https://github.com/yyxd/leetcode/blob/master/src/leetcode/strings/Problem344_ReverseString.java

2. Reverse Words in a String

java

StringBuilder

split

https://github.com/yyxd/leetcode/blob/master/src/leetcode/strings/Problem151_ReverseWordsInString.java

3. String to Integer (atoi)

(atoi)

,

+/-

break

INT.MAX

INT.MAX, INT.MIN

INT.MIN

long

int

https://github.com/yyxd/leetcode/blob/master/src/leetcode/strings/Problem8_atoi.java

- 2019-02-08 23:13:52

itoa

```

public class Solution {
    public int myAtoi(String str) {
        if (str.isEmpty())
            return 0;
        str = str.trim();
        int i = 0, ans = 0, sign = 1, len = str.length();
        if (str.charAt(i) == '-' || str.charAt(i) == '+')
            sign = str.charAt(i++) == '+' ? 1 : -1;
        for (; i < len; ++i) {
            int tmp = str.charAt(i) - '0';
            if (tmp < 0 || tmp > 9)
                break;
            if (ans > Integer.MAX_VALUE / 10
                || (ans == Integer.MAX_VALUE / 10 && Integer.MAX_VALUE % 10 < tmp))
                return sign == 1 ? Integer.MAX_VALUE : Integer.MIN_VALUE;
            else
                ans = ans * 10 + tmp;
        }
    }
}

```

+ 8Um{

```
return sign * ans;
}
}
```

- _CountingStars 2019-02-08 22:02:57

```
go
package main

import "fmt"

func reverseString(s []byte) {
length := len(s)
for i := 0; i < length/2; i++ {
s[i], s[length-i-1] = s[length-i-1], s[i]
}
}

func main() {
testString := []byte{'h', 'e', 'l', 'l', 'o'}
fmt.Println(string(testString[:]))
reverseString(testString)
fmt.Println(string(testString[:]))
}
```

- 2019-02-08 21:41:25

```
LRU
private class Node{
private Node prev;
private Node next;
private int key;
private int value;
```

```
Node(int key,int value){
```


+

8Um{

this.key = key;

this.value = value;

}

}

private Node head;//

private Node tail;//

private Map<Integer,Node> cache;

private int capacity;

public LRUCache(int capacity) {

this.cache = new HashMap<>();

this.capacity = capacity;

}

public int get(int key) {

Node node = cache.get(key);

if(node == null){

return -1;

}else{

moveNode(node);

return node.value;

}

}

public void put(int key, int value) {

Node node = cache.get(key);

if (node != null){

node.value = value;

moveNode(node);

}else {

removeHead();

Z`Y.###>#l YY_ha Y#

E E %\$+\$\$&- +#YVcc_#

#

+

8Um{

"\ha `Q&\$% #&#%`&% +.) - Q

+

private

```

        addNode(new Node(key,value));
    }
    cache.put(key,node);
}

private void removeHead(){
    if (cache.size() == capacity){
        Node tempNode = head;
        cache.remove(head.key);
        head = head.next;
        tempNode.next = null;
        if (head != null)
            head.prev = null;
    }
}

private void addNode(Node node){
    if (head == null)
        head = tail = node;
    else
        addNodeToTail(node);
}

private void addNodeToTail(Node node){
    node.prev = tail;
    tail.next = node;
    tail = node;
}

private void moveNode(Node node){
    if(head == node && node != tail){
        head = node.next;
    }
}

```

+

8Um{

```
head.prev = null;
node.next = null;
addNodeToTail(node);
}else if (tail == node){
}else {
node.prev.next = node.next;
node.next.prev = node.prev;
node.next = null;
addNodeToTail(node);
}
}
}
```

- 2019-02-08 19:33:04

```
class Solution {
public void reverseString(char[] s) {
int start = 0;
int end = s.length - 1;
while(start < end){
swap(s,start,end);
start++;
end--;
}
}
```

```
public void swap(char[] array,int a,int b){
char tmp = array[a];
array[a] = array[b];
array[b] = tmp;
}
```

}

- 2019-02-08 18:23:42

//

package com.jxyang.test.geek.day4.Solution;

class Solution2 {

public int myAtoi(String str) {

if(str==null){

return 0;

}

char[] arr= str.toCharArray();

boolean flag = false;

boolean numBegin = false;

int result = 0;

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

if(numBegin && (arr[i]=='-'||arr[i]=='+'||arr[i]==' ')){

break;

}else if(arr[i]==' ') {

continue;

}else if(arr[i]=='+'){

numBegin = true;

continue;

}else if(arr[i]=='-'){

flag = true;

numBegin = true;

continue;

}else if(arr[i]>='0'&&arr[i]<='9'){

numBegin = true;

if(result==0){

result = flag?('0'-arr[i]):(arr[i]-'0');

}else{

+

8Un{

```
try{
result = Math.multiplyExact(result,10);
result = Math.addExact(result,flag?('0'-arr[i]):(arr[i]-'0'));
}catch (Exception e){
if(flag){
return Integer.MIN_VALUE;
}else{
return Integer.MAX_VALUE;
}
}
}
}
}else{
break;
}
}
return result;
}

public static void main(String[] args) {
Solution2 solution2 = new Solution2();
System.out.println(solution2.myAtoi("42"));
System.out.println(solution2.myAtoi(" +0 123"));//      123
System.out.println(solution2.myAtoi(" -42"));
System.out.println(solution2.myAtoi("4193 with words"));
System.out.println(solution2.myAtoi("words and 987"));
System.out.println(solution2.myAtoi("-91283472332"));//      -2147483648
System.out.println(solution2.myAtoi("+1"));//      -2147483648
}
}
```

- 2019-02-08 17:12:16

```
class Solution {
```

```
//
```

+

8Um{

```
public void reverseString(char[] s) {  
    if(s==null||s.length<2){  
        return;  
    }  
    int l=0;  
    int r=s.length-1;  
    while (l<r){  
        char tmp = s[l];  
        s[l] = s[r];  
        s[r] = tmp;  
        l++;  
        r--;  
    }  
}
```

- C_love 2019-02-08 12:40:05

Reverse Words in a String

```
public class Solution {  
    public String reverseWords(String s) {  
        final List<String> words = new ArrayList<>();  
        final char[] charArray = s.toCharArray();  
  
        int start = 0;  
        int end = 0;  
        while (end < s.length()) {  
            if (' ' == charArray[end]) {  
                if (start != end) {  
                    words.add(getWord(charArray, start, end));  
                    start = end;  
                }  
            }  
            end++;  
        }  
        if (start != end) {  
            words.add(getWord(charArray, start, end));  
        }  
        return String.join(" ", words);  
    }  
}
```

+

8Um{

start++;

end++;

} else {

end++;

}

}

if (start != end) {

words.add(getWord(charArray, start, end));

}

Collections.reverse(words);

return String.join(" ", words);

}

private String getWord(final char[] charArray, final int start, final int end) {

char[] tmp = new char[end - start];

int pos = 0;

for(int i = start; i < end; i++) {

tmp[pos++] = charArray[i];

}

return new String(tmp);

}

}