Previous (/articles/loud-and-rich/) Next (/articles/reverse-integer/)

6. ZigZag Conversion (/problems/zigzag-conversion/)

June 11, 2018 | 79.3K views

Average Rating: 4.14 (42 votes)

The string `"PAYPALISHIRING"` is written in a zigzag pattern on a given number of rows like this: (you may want to display this pattern in a fixed font for better legibility)

```
P   A   H   N
A P L S I I G
Y   I   R
```

And then read line by line: `"PAHNAPLSIIGYIR"`

Write the code that will take a string and make this conversion given a number of rows:

```
string convert(string s, int numRows);
```

Example 1:

```
Input: s = "PAYPALISHIRING", numRows = 3
Output: "PAHNAPLSIIGYIR"
```

Example 2:

```
Input: s = "PAYPALISHIRING", numRows = 4
Output: "PINALSIGYAHRPI"
Explanation:

P     I    N
A   L S  I G
Y A   H R
P     I
```

Solution

Approach 1: Sort by Row

Intuition

By iterating through the string from left to right, we can easily determine which row in the Zig-Zag pattern that a character belongs to.

Algorithm

We can use $\min(\text{numRows}, \text{len}(s))$ lists to represent the non-empty rows of the Zig-Zag Pattern.

Iterate through s from left to right, appending each character to the appropriate row. The appropriate row can be tracked using two variables: the current row and the current direction.

The current direction changes only when we moved up to the topmost row or moved down to the bottommost row.

C++ Java Copy

```cpp
class Solution {
public:
    string convert(string s, int numRows) {

        if (numRows == 1) return s;

        vector<string> rows(min(numRows, int(s.size())));
        int curRow = 0;
        bool goingDown = false;

        for (char c : s) {
            rows[curRow] += c;
            if (curRow == 0 || curRow == numRows - 1) goingDown = !goingDown;
            curRow += goingDown ? 1 : -1;
        }

        string ret;
        for (string row : rows) ret += row;
        return ret;
    }
};
```

Complexity Analysis

- Time Complexity: $O(n)$, where $n == \text{len}(s)$
- Space Complexity: $O(n)$

Approach 2: Visit by Row

Intuition

Visit the characters in the same order as reading the Zig-Zag pattern line by line.

Algorithm

Visit all characters in row 0 first, then row 1, then row 2, and so on...

For all whole numbers k,

- Characters in row 0 are located at indexes $k\ (2 \cdot \text{numRows} - 2)$
- Characters in row $\text{numRows} - 1$ are located at indexes $k\ (2 \cdot \text{numRows} - 2) + \text{numRows} - 1$

- Characters in inner row i are located at indexes $k\ (2 \cdot \text{numRows} - 2) + i$ and $(k+1)(2 \cdot \text{numRows} - 2) - i$.

C++ Java Copy

```
class Solution {
public:
    string convert(string s, int numRows) {

        if (numRows == 1) return s;

        string ret;
        int n = s.size();
        int cycleLen = 2 * numRows - 2;

        for (int i = 0; i < numRows; i++) {
            for (int j = 0; j + i < n; j += cycleLen) {
                ret += s[j + i];
                if (i != 0 && i != numRows - 1 && j + cycleLen - i < n)
                    ret += s[j + cycleLen - i];
            }
        }
        return ret;
    }
};
```

Complexity Analysis

- Time Complexity: $O(n)$, where $n == \text{len}(s)$. Each index is visited once.
- Space Complexity: $O(n)$. For the cpp implementation, $O(1)$ if return string is not considered extra space.

Rate this article:

Previous (/articles/loud-and-rich/) Next (/articles/reverse-integer/)

Comments: 62

Sort By

Type comment here... (Markdown is supported)

Preview Post

cptdav (cptdav) ★ 13 August 16, 2018 12:28 PM

a python solution using O(1) space and O(n) time:

```
class Solution(object):
    def convert(self, s, numRows):
        if s == None:
```

Read More

12 Share Reply

SHOW 2 REPLIES

lwxiaolei (lwxiaolei) ★ 24 August 13, 2018 2:24 AM

python solution:

```
class Solution(object):
    def convert(self, s, numRows):
        if numRows == 1:
```

Read More

10 Share Reply

h11129 (h11129) ★ 8 October 3, 2018 11:11 PM

I think the solution is actually not zigzag as what's told in the problem. It work because we only need to print out the zigzag line by line. It won't work if the question is to print out the zigzag. It lose the shape.

6 Share Reply

NaT3z (nat3z) ★ 11 October 23, 2018 9:09 AM

Simple Python3 solution using dict/hashtable. I like it because it's intuitive and the "step" between rows can be visualised. 76ms runtime (99.62th percentile).

try except block without specific error (KeyError to avoid string length < numRows) might be bad form but specifying strangely increased runtime considerably.

Read More

6 Share Reply

user7579F (user7579f) ★ 5 January 11, 2019 2:29 PM

Python 3 solution using a generator:

Read More

5 Share Reply

Ronnie268 (ronnie268) ★ 4 December 23, 2018 3:23 PM

Python3 (beats 95%):

```
class Solution:
    def convert(self, s, numRows):
```

Read More

4 Share Reply

humy (humy) ★ 15 January 21, 2019 11:19 AM

o(n): 80 ms, faster than 99.72% of Python3
when we travels a v shape in the z shape of zigzag,we finish a loop of all the change in row number.

```
class Solution:
    def convert(self, s, numRows):
```

Read More

4 Share Reply

wguo32 (wguo32) ★ 16 September 16, 2018 2:29 PM

my clumsy answer:

Read More

5 Share Reply

mrooo (mrooo) ★ 3 September 24, 2018 7:36 AM

Read More

3 Share Reply

zkf85 (zkf85) ★ 9 March 27, 2019 8:49 PM

Python Solution:

Read More

3 Share Reply

1 2 3 4 5 6 7