ZigZag Conversion

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 text, int nRows);
```

`convert("PAYPALISHIRING", 3)` should return `"PAHNAPLSIIGYIR"`.

Solution 1

Create nRows StringBuffers, and keep collecting characters from original string to corresponding StringBuffer. Just take care of your index to keep them in bound.

```
public String convert(String s, int nRows) {
    char[] c = s.toCharArray();
    int len = c.length;
    StringBuffer[] sb = new StringBuffer[nRows];
    for (int i = 0; i < sb.length; i++) sb[i] = new StringBuffer();

    int i = 0;
    while (i < len) {
        for (int idx = 0; idx < nRows && i < len; idx++) // vertically down
            sb[idx].append(c[i++]);
        for (int idx = nRows-2; idx >= 1 && i < len; idx--) // obliquely up
            sb[idx].append(c[i++]);
    }
    for (int idx = 1; idx < sb.length; idx++)
        sb[0].append(sb[idx]);
    return sb[0].toString();
}
```

written by dylan_yu original link here

Solution 2

```
/*n=numRows
Δ=2n-2    1                           2n-1                          4n-3
Δ=        2                     2n-2  2n                    4n-4    4n-2
Δ=        3               2n-3        2n+1              4n-5        .
Δ=        .           .               .                 .           .
Δ=        .       n+2                 .           3n                .
Δ=        n-1 n+1                     3n-3    3n-1                  5n-5
Δ=2n-2    n                           3n-2                          5n-4
*/
```

that's the zigzag pattern the question asked! Be careful with nR=1 && nR=2

my 16ms code in c++:

```
class Solution {
public:
    string convert(string s, int numRows) {
        string result="";
        if(numRows==1)
            return s;
        int step1,step2;
        int len=s.size();
        for(int i=0;i<numRows;++i){
            step1=(numRows-i-1)*2;
            step2=(i)*2;
            int pos=i;
            if(pos<len)
                result+=s.at(pos);
            while(1){
                pos+=step1;
                if(pos>=len)
                    break;
                if(step1)
                    result+=s.at(pos);
                pos+=step2;
                if(pos>=len)
                    break;
                if(step2)
                    result+=s.at(pos);
            }
        }
        return result;
    }
};
```

written by HelloKenLee original link here

Solution 3

The problem statement itself is unclear for many. Especially for 2-row case. "ABCD", 2 --> "ACBD". The confusion most likely is from the character placement. I would like to extend it a little bit to make ZigZag easy understood.

The example can be written as follow:

1. P.....A.....H.....N
2. ..A..P....L..S....I...I....G
3. ....Y.....I.....R

Therefore, <ABCD, 2> can be arranged as:

1. A....C
2. ...B....D

My simple accepted code:

```
string convert(string s, int nRows) {

    if (nRows <= 1)
        return s;

    const int len = (int)s.length();
    string *str = new string[nRows];

    int row = 0, step = 1;
    for (int i = 0; i < len; ++i)
    {
        str[row].push_back(s[i]);

        if (row == 0)
            step = 1;
        else if (row == nRows - 1)
            step = -1;

        row += step;
    }

    s.clear();
    for (int j = 0; j < nRows; ++j)
    {
        s.append(str[j]);
    }

    delete[] str;
    return s;
}
```

written by enze98 original link here