ZigZag Conversion

```
class Solution:
    def convert(self, s: str, numRows: int) → str:
        #Initial thoughts is to use a list of arrays, have it so initial words are insert
ed at first position of each
        #array and when it hits the row limit,start moving diagonally, (move down
until hit row limit, move right up, insert, right up insert until hit row limit, and th
en move down again)
        #repeat until strings are exhausted
        rows = [[] for _ in range(numRows)] #Creates rows of lists
        row = 0 #Tracks current row
        dr = +1 #Controls direction
        if numRows == 1:
            return s

        for char in s:
         rows[row].append(char)

         if row == 0:
            dr = 1
         elif row == numRows -1: #Whenever it hits a boundary, flip direction app
end upwards
            dr = -1
         row += dr
        return "".join("".join(r) for r in rows)#You don't actually need to worry abo
ut the column row, you can just map out the rows of where the string is
    #Dont forget to iteratively join the inner rows and then the outer row
```

- Overall a good question, I did manage to break it down enough to understand that the movements of a ZigZag had to be tracked, I was concerned with column positions but I didn’t actually have to use it.

- All you had to do was flip the direction of appending rows for every boundary hit