8. String to Integer (atoi)

Question | My Submissions

Total Accepted: **79793** Total Submissions: **609124** Difficulty: **Easy**

> Implement atoi to convert a string to an integer.
>
> **Hint:** Carefully consider all possible input cases. If you want a challenge, please do not see below and ask yourself what are the possible input cases.
>
> **Notes:** It is intended for this problem to be specified vaguely (ie, no given input specs). You are responsible to gather all the input requirements up front.

The atoi function first discards as many whitespace characters as necessary until the first non-whitespace character is found.

Then, starting from this character, takes an optional initial plus or minus sign followed by as many numerical digits as possible, and interprets them as a numerical value.

The string can contain additional characters after those that form the integral number, which are ignored and have no effect on the behavior of this function.

If the first sequence of non-whitespace characters in str is not a valid integral number, or if no such sequence exists because either str is empty or it contains only whitespace characters, no conversion is performed.

If no valid conversion could be performed, a zero value is returned.

step1. remove all empty space in the beginning and the end of the string
step2. define a flag to mark the positive and negative sign
step3. define an index (like a pointer) for looping through the string
step4. looping

case 1: if the char is digit and we are not at the edge of the string
result = result * 10 + char

case 2: if we are at the edge of the string and the result is greater than 214748364 (meaning that the number will overflow)
just return 2147483647 (max positive) or -2147483648 (min negative) depends on the flag

case 3: if the char is not digit, there is no way to convert the string to number, break the loop

step5. return result times the flag,

```python
class Solution:
    def myAtoi(slef, s):

        s = s.strip()  # remove the whitespace
        n = len(s)
        i = 0          # index use to keep track on whihc character we are dealing with
        flag=1         # negative or positive
        result = 0     # store the result
        maxint = 2147483647
        minint = -2147483648
        aboutOverflow = 214748364


        # we need to determine the sign or the number first, then we can analysis the rest
        # of the string
        if i < n and s[i] == '+':
            i += 1
        elif i < n and s[i] == '-':
            i += 1
            flag = -1

        while i < n:
            if s[i].isdigit():
                # if s[i] is digit, see if it is going to oveflow first.
                # if it is about to overflow, just return the maximum int
                if result > aboutOverflow or (result == aboutOverflow and s[i] >= 8):
                    if flag = 1:
                        return maxint
                    else:
                        return minint
                else:  # if its not about to overflow, accumulate the result
                    result = result * 10 + int(s[i])
                i += 1
            else:# if is not digit, break the loop
                break

        return result*flag
```