Problem | Editorial | Submissions

Implement Atoi

Difficulty: **Medium** Accuracy: **32.58%** Submissions: **246K+** Points: **4**

Given a string s, the objective is to convert it into integer format without utilizing any built-in functions. Refer the below steps to know about atoi() function.

Cases for atoi() conversion:

1. Skip any leading whitespaces.
2. Check for a sign ('+' or '-'), default to positive if no sign is present.
3. Read the integer by ignoring leading zeros until a non-digit character is encountered or end of the string is reached. If no digits are present, return 0.
4. If the integer is greater than $2^{31} - 1$, then return $2^{31} - 1$ and if the integer is smaller than -2^{31}, then return -2^{31}.

Examples:

Input: s = "-123"
Output: -123
Explanation: It is possible to convert -123 into an integer so we returned in the form of an integer

Java (1.8) | Average Time: 15m | Start Timer

```java
// } Driver Code Ends

class Solution {
    public int myAtoi(String s) {
        int i = 0, n = s.length();
        while (i < n && s.charAt(i) == ' ') {
            i++;
        }
        if (i == n) return 0;
        int sign = 1;
        if (s.charAt(i) == '-') {
            sign = -1;
            i++;
        } else if (s.charAt(i) == '+') {
            i++;
        }
        long result = 0;
        while (i < n && Character.isDigit(s.charAt(i))) {
            result = result * 10 + (s.charAt(i) - '0');
            if (result * sign > Integer.MAX_VALUE) return Integer.MAX_VALUE;
            if (result * sign < Integer.MIN_VALUE) return Integer.MIN_VALUE;
            i++;
        }
        return (int) (result * sign);
    }
}
```

Custom Input | Compile & Run | Submit