

Implement Atoi

🔽 solved by	Senan
🔽 Platform	GeeksForGeeks
🔗 difficulty	Medium
≡ tags	LogicString Manipulation
🗨 language	C++
📅 solved on	@28/11/2024
🔗 link	https://www.geeksforgeeks.org/problems/implement-atoi/1
☑ Completion	✓

Intuition

The problem is to convert a string into an integer (similar to the `atoi` function in C). The idea is to:

1. Skip any leading whitespace.
2. Check for an optional sign ('+' or '-').
3. Process the numeric characters while ensuring the result remains within the valid integer range.
4. Return the result as an integer, clamping it to `INT_MAX` or `INT_MIN` if overflow occurs.

Approach

1. **Skip Whitespaces:** Traverse the input string to ignore leading spaces.
2. **Handle Sign:** Check if the first non-space character is a '+' or '-', determining the sign of the result.
3. **Build the Number:** Iterate through the characters, converting each valid digit to its integer form and updating the result.
4. **Check Overflow:** Compare the accumulated result with `INT_MAX` or `INT_MIN` at each step to avoid overflow.
5. **Return Result:** Return the final value, adjusted by the sign.

Complexity

Time Complexity:

- **$O(n)$:** The algorithm scans the string once, where `n` is the length of the input string.

Space Complexity:

- **$O(1)$:** Only a few integer variables are used for processing, and no additional data structures are required.

Code

```
class Solution {
public:
```

```
int myAtoi(char *s) {
    int i = 0, sign = 1;
    long result = 0;

    while (s[i] == ' ') {
        i++;
    }

    if (s[i] == '+' || s[i] == '-') {
        sign = (s[i] == '-') ? -1 : 1;
        i++;
    }

    while (s[i] >= '0' && s[i] <= '9') {
        result = result * 10 + (s[i] - '0');

        if (sign * result >= INT_MAX) return INT_MAX;
        if (sign * result <= INT_MIN) return INT_MIN;
        i++;
    }

    return (int)(sign * result);
};
```