

# Assignment Week 2

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

Screenshot of a LeetCode submission page for the problem "String to Integer (atoi)".

The page shows the problem description, a list of tabs (Description, Accepted, Editorial, Solutions, Submissions, Testcase, Code), and a submission by user "arnabdas1999" submitted on Mar 11, 2025 at 20:59. The submission is marked as "Accepted".

The runtime performance is shown as 1 ms, Beats 100.00%. The memory usage is 42.44 MB, 64.66%.

A bar chart shows the runtime performance across different test cases, with the first case being the most time-consuming.

The code editor shows the following Java code:

```
1 class Solution {
2     public int myAtoi(String s) {
3         int i=0,sign=1,n=s.length(),result =0;
4         while(i<n && s.charAt(i)==' '){
5             i++;
6         }
7         if(i<n && (s.charAt(i) == '-' || s.charAt(i) == '+')){
8             sign = (s.charAt(i)=='-')? -1:1;
9             i++;
10        }
11        while(i<n && Character.isDigit(s.charAt(i))){
12            int digit = s.charAt(i) - '0';
13            if(result>(Integer.MAX_VALUE-digit)/10){
14                return (sign == 1)? Integer.MAX_VALUE:Integer.MIN_VALUE;
15            }
16            result = result*10+digit;
17            i++;
18        }
19        return sign*result;
20    }
21 }
```

The test result shows "Accepted" with a runtime of 0 ms. The input is "s = \"42\"", and the output is "42".

## Question 2

```
class Solution {
    public List<Integer> findAnagrams(String s, String p) {
        int [] plen = new int[26];
        int [] slen = new int[26];
        List<Integer> result = new ArrayList<>();
        for(int i = 0; i < p.length(); i++){
            plen[p.charAt(i) - 'a']++;
        }
        int plength = p.length();
        for(int i = 0; i < s.length(); i++){
            slen[s.charAt(i) - 'a']++;
            if(i >= plength){
                slen[s.charAt(i-plength) - 'a']--;
            }
            if(Arrays.equals(plen, slen)){
                result.add(i-plength+1);
            }
        }
        return result;
    }
}
```

leetcode.com/problems/find-all-anagrams-in-a-string/

Problem List < > 🔍

Description Accepted Editorial Solutions Submissions Testcase </> Code

All Submissions

Accepted 100% 100 submissions passed

arnabdas1999 submitted at Mar 11, 2025 21:02

Runtime 10 ms Beats 79.11%

Analyze Complexity

Memory 44.68 MB 91.10%

11ms 443ms 875ms 1307ms

Editorial Solution

```
1 class Solution {
2     public List<Integer> findAnagrams(String s, String p) {
3         int [] plen = new int[26];
4         int [] slen = new int[26];
5         List<Integer> result = new ArrayList<>();
6         for(int i = 0; i < p.length(); i++){
7             plen[p.charAt(i) - 'a']++;
8         }
9         int plength = p.length();
10        for(int i = 0; i < s.length(); i++){
11            slen[s.charAt(i) - 'a']++;
12            if(i >= plength){
13                slen[s.charAt(i-plength) - 'a']--;
14            }
15            if(Arrays.equals(plen, slen)){
16                result.add(i-plength+1);
17            }
18        }
19        return result;
20    }
21 }
```

Go 25.00s 2.00s

Run Submit

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

```
class Solution {
    public List<Integer> findAnagrams(String s, String p) {
```

### Question 3

```
class Solution {  
  
    public String reverseWords(String s) {  
        String[] words= s.split("\\s+");  
        StringBuilder res = new StringBuilder();  
        for(int i = words.length -1;i>=0;i--){  
            res.append(words[i]);  
            if(i!=0){  
                res.append(" ");  
            }  
        }  
        return res.toString().trim();  
    }  
}
```

The screenshot shows a LeetCode submission page for the problem "Reverse Words in a String". The submission is accepted, with a runtime of 6 ms, beating 85.99% of other submissions. The memory usage is 43.54 MB, which is 37.94% of the limit. A bar chart shows the distribution of runtime times, with a peak at 5ms. The code is written in Java and is displayed in the editor. The test results show that the solution passes all three test cases.

Accepted 100% 10 testcases passed  
arnabdas1999 submitted at Mar 11, 2025 21:03

Runtime: 6 ms Beats 85.99%  
Memory: 43.54 MB 37.94%

Editorial Solution

```
1 class Solution {  
2     public String reverseWords(String s) {  
3         String[] words= s.split("\\s+");  
4         StringBuilder res = new StringBuilder();  
5         for(int i = words.length -1;i>=0;i--){  
6             res.append(words[i]);  
7             if(i!=0){  
8                 res.append(" ");  
9             }  
10        }  
11        return res.toString().trim();  
12    }  
13 }  
14 }
```

Test Result: Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input: s = "the sky is blue"

Output: "blue is sky the"