

## 942. DI String Match

Solved

Easy

Topics

Companies

A permutation `perm` of  $n + 1$  integers of all the integers in the range  $[0, n]$  can be represented as a string `s` of length `n` where:

- `s[i] == 'I'` if `perm[i] < perm[i + 1]`, and
- `s[i] == 'D'` if `perm[i] > perm[i + 1]`.

Given a string `s`, reconstruct the permutation `perm` and return it. If there are multiple valid permutations `perm`, return **any of them**.

### Example 1:

Input: `s = "IDID"`

Output: `[0,4,1,3,2]`

### Example 2:

Input: `s = "III"`

Output: `[0,1,2,3]`

### Example 3:

Input: `s = "DDI"`

Output: `[3,2,0,1]`

Problem List | Run | Submit | Premium

Description | Accepted | Editorial | Solutions | Submissions

← All Submissions

Accepted

Zhukovilya submitted at Jun 30, 2024 17:46

Editorial | Solution

Runtime: 96 ms | Beats 76.98% | Memory: 50.70 MB | Beats 65.87%

Analyze Complexity

0.79% of solutions used 117 ms of runtime

Code | C#

```
public class Solution {
    public int[] DiStringMatch(string s) {
        int n = s.Length;
        int[] result = new int[n + 1];
        int low = 0;
        int high = n;

        for (int i = 0; i < n; i++) {
            if (s[i] == 'I') {
                result[i] = low++;
            } else {
                result[i] = high--;
            }
        }
        result[n] = low;
        return result;
    }
}
```

Testcase | Test Result

Case 1 | Case 2 | Case 3 | +

s = "IDID"

</> Source

Код:

```
public class Solution
{
    public int[] DiStringMatch(string s)
    {
        int n = s.Length;
        int[] result = new int[n + 1];
```

```
int low = 0;
int high = n;

for (int i = 0; i < n; i++)
{
    if (s[i] == 'I')
    {
        result[i] = low++;
    }
    else
    {
        result[i] = high--;
    }
}
result[n] = low; // Последний элемент будет равен оставшемуся значению
return result;
}
```