

929. Unique Email Addresses

Solved

Easy

Topics

Companies

Every **valid email** consists of a **local name** and a **domain name**, separated by the '@' sign. Besides lowercase letters, the email may contain one or more '.' or '+'.

- For example, in "alice@leetcode.com", "alice" is the **local name**, and "leetcode.com" is the **domain name**.

If you add periods '.' between some characters in the **local name** part of an email address, mail sent there will be forwarded to the same address without dots in the local name. Note that this rule **does not apply to domain names**.

- For example, "alice.z@leetcode.com" and "alicez@leetcode.com" forward to the same email address.

If you add a plus '+' in the **local name**, everything after the first plus sign **will be ignored**. This allows certain emails to be filtered. Note that this rule **does not apply to domain names**.

- For example, "m.y+name@email.com" will be forwarded to "my@email.com".

It is possible to use both of these rules at the same time.

Given an array of strings `emails` where we send one email to each `emails[i]`, return the number of different addresses that actually receive mails.

The screenshot shows the LeetCode interface for problem 929. On the left, the 'All Submissions' tab is active, displaying a bar chart of runtime performance. The chart shows a distribution of runtimes, with a peak around 71ms. Below the chart, the C# code for the solution is visible. On the right, the 'Code' tab is active, showing the C# code for the solution. The code uses a HashSet to store unique email addresses after applying the rules for dots and plus signs. The 'Testcase' tab shows the input and output for the test cases.

Runtime Performance:

- Runtime: 71 ms | Beats: 49.24%
- Memory: 45.83 MB | Beats: 73.66%

Code:

```
public class Solution {
    public int NumUniqueEmails(string[] emails) {
        HashSet<string> uniqueEmails = new HashSet<string>();
        foreach (string email in emails) {
            string[] parts = email.Split('@');
            string local = parts[0];
            string domain = parts[1];
            // Удаление части после '+'
            int plusIndex = local.IndexOf('+');
            if (plusIndex != -1) {
                local = local.Substring(0, plusIndex);
            }
            local = local.Replace(".", "");
            uniqueEmails.Add(local + domain);
        }
        return uniqueEmails.Count;
    }
}
```

Testcase:

Input: ["test.email+alex@leetcode.com", "test.e.mail+bob.cathy@leetcode.com", "testemail+ david@lee.tcode.com"]

Output: 2

Код:

```
using System.Collections.Generic;

public class Solution
{
    public int NumUniqueEmails(string[] emails)
    {
        HashSet<string> uniqueEmails = new HashSet<string>();
```

```

foreach (string email in emails)
{
    string[] parts = email.Split('@');
    string local = parts[0];
    string domain = parts[1];

    // Удаление части после '+'
    int plusIndex = local.IndexOf('+');
    if (plusIndex != -1)
    {
        local = local.Substring(0, plusIndex);
    }

    // Удаление всех точек
    local = local.Replace(".", "");

    // Создание нормализованного адреса электронной почты
    string normalizedEmail = local + "@" + domain;

    // Добавление нормализованного адреса в хэш-таблицу
    uniqueEmails.Add(normalizedEmail);
}

return uniqueEmails.Count;
}
}

```