```
class Solution {
    public int numUniqueEmails(String[] emails) {
        Set<String> uniqueEmails = new HashSet<>();
                for (String email : emails) {
                        String[] parts = email.split("@");
                        StringBuilder sb = new StringBuilder();
                        String localName = parts[0];
                        for (int i = 0; i < localName.length(); i++) {
                                char c = localName.charAt(i);
                                if (c == '+')
                                        break;
                                if (c != '.')
                                        sb.append(c);
                        }
                        uniqueEmails.add(sb.append("@").append(parts[1]).toString());
                }
                return uniqueEmails.size();
    }
}
```

Every email consists of a local name and a domain name, separated by the @ sign.

For example, in `alice@leetcode.com`, `alice` is the local name, and `leetcode.com` is the domain name.

Besides lowercase letters, these emails may contain `'.'`s or `'+'`s.

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. For example, `"alice.z@leetcode.com"` and `"alicez@leetcode.com"` forwa rd to the same email address. (Note that this rule does not apply for domain names.)

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, for example `m.y+name@email.com` will be forwarded to `my@email.com`. (Again, this rule does not apply for domain names.)

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

Given a list of `emails`, we send one email to each address in the list. How many different addresses actually receive mails?