

具体请参看如下代码中的 `maxUnique` 方法。

```
public int maxUnique(String str) {
    if (str == null || str.equals("")) {
        return 0;
    }
    char[] chas = str.toCharArray();
    int[] map = new int[256];
    for (int i = 0; i < 256; i++) {
        map[i] = -1;
    }
    int len = 0;
    int pre = -1;
    int cur = 0;
    for (int i = 0; i != chas.length; i++) {
        pre = Math.max(pre, map[chas[i]]);
        cur = i - pre;
        len = Math.max(len, cur);
        map[chas[i]] = i;
    }
    return len;
}
```

## 找到被指的新类型字符

### 【题目】

新类型字符的定义如下：

1. 新类型字符是长度为 1 或者 2 的字符串。
2. 表现形式可以仅是小写字母，例如，"e"；也可以是大写字母+小写字母，例如，"Ab"；还可以是大写字母+大写字母，例如，"DC"。

现在给定一个字符串 `str`，`str` 一定是若干新类型字符正确组合的结果。比如"eaCCBi"，由新类型字符"e"、"a"、"CC"和"Bi"拼成。再给定一个整数 `k`，代表 `str` 中的位置。请返回被 `k` 位置指中的新类型字符。

### 【举例】

`str="aaABCDEcBCg"`。

1. `k=7` 时，返回"Ec"。
2. `k=4` 时，返回"CD"。
3. `k=10` 时，返回"g"。

## 【难度】

士 ★☆☆☆

## 【解答】

一种笨方法是从 `str[0]` 开始，从左到右依次划分出新类型字符，到 `k` 位置的时候就知道指向的新类型字符是什么。比如 `str="aaABCDEcBCg"`，`k=7`。从左到右可以依次划分出"a"、"a"、"AB"、"CD"。然后发现 `str[7]` 是大写字母'E'，所以被指中的新类型字符一定是"EC"，返回即可。

更快的方法。从 `k-1` 位置开始，向左统计连续出现的大写字母的数量记为 `uNum`，遇到小写字母就停止。如果 `uNum` 为奇数，`str[k-1..k]` 是被指中的新类型字符，见例子 1。如果 `uNum` 为偶数且 `str[k]` 是大写字母，`str[k..k+1]` 是被指中的新类型字符，见例子 2。如果 `uNum` 为偶数且 `str[k]` 是小写字母，`str[k]` 是被指中的新类型字符，见例子 3。

具体过程请参看如下代码中的 `pointNewchar` 方法

```
public String pointNewchar(String s, int k) {
    if (s == null || s.equals("") || k < 0 || k >= s.length()) {
        return "";
    }
    char[] chas = s.toCharArray();
    int uNum = 0;
    for (int i = k - 1; i >= 0; i--) {
        if (!isUpper(chas[i])) {
            break;
        }
        uNum++;
    }
    if ((uNum & 1) == 1) {
        return s.substring(k - 1, k + 1);
    }
    if (isUpper(chas[k])) {
        return s.substring(k, k + 2);
    }
    return String.valueOf(chas[k]);
}
```