```
        int mid = 0;
        int i = 0;
        while (left <= right) {
            mid = (left + right) / 2;
            if (strs[mid] != null && strs[mid].equals(str)) {
                res = mid;
                right = mid - 1;
            } else if (strs[mid] != null) {
                if (strs[mid].compareTo(str) < 0) {
                    left = mid + 1;
                } else {
                    right = mid - 1;
                }
            } else {
                i = mid;
                while (strs[i] == null && --i >= left)
                    ;
                if (i < left || strs[i].compareTo(str) < 0) {
                    left = mid + 1;
                } else {
                    res = strs[i].equals(str) ? i : res;
                    right = i - 1;
                }
            }
        }
        return res;
    }
```

字符串的调整与替换

【题目】

给定一个字符类型的数组 chas[]，chas 右半区全是空字符，左半区不含有空字符。现在想将左半区中所有的空格字符替换成"%20"，假设 chas 右半区足够大，可以满足替换所需要的空间，请完成替换函数。

【举例】

如果把 chas 的左半区看作字符串，为"a b c"，假设 chas 的右半区足够大。替换后，chas 的左半区为"a%20b%20%20c"。

【要求】

替换函数的时间复杂度为 $O(N)$，额外空间复杂度为 $O(1)$。

【补充题目】

给定一个字符类型的数组 chas[]，其中只含有数字字符和“*”字符。现在想把所有的“*”字符挪到 chas 的左边，数字字符挪到 chas 的右边。请完成调整函数。

【举例】

如果把 chas 看作字符串，为"12**345"。调整后 chas 为"**12345"。

【要求】

1. 调整函数的时间复杂度为 $O(N)$，额外空间复杂度为 $O(1)$。
2. 不得改变数字字符从左到右出现的顺序。

【难度】

士 ★☆☆☆

【解答】

原问题。遍历一遍可以得到两个信息，chas 的左半区有多大，记为 len，左半区的空格数有多少，记为 num，那么可知空格字符被“%20”替代后，长度将是 len+2*num。接下来从左半区的最后一个字符开始倒着遍历，同时将字符复制到新长度最后的位置，并依次向左倒着复制。遇到空格字符就依次把“0”、“2”和“%”进行复制。这样就可以得到替换后的 chas 数组。具体过程请参看如下代码中的 replace 方法。

```
public void replace(char[] chas) {
        if (chas == null || chas.length == 0) {
                return;
        }
        int num = 0;
        int len = 0;
        for (len = 0; len < chas.length && chas[len] != 0; len++) {
                if (chas[len] == ' ') {
                        num++;
                }
        }
        int j = len + num * 2 - 1;
        for (int i = len - 1; i > -1; i--) {
                if (chas[i] != ' ') {
                        chas[j--] = chas[i];
                } else {
                        chas[j--] = '0';
```

```
                        chas[j--] = '2';
                        chas[j--] = '%';
                }
        }
}
```

补充问题。依然是从右向左倒着复制，遇到数字字符则直接复制，遇到“*”字符不复制。当把数字字符复制完，把左半区全部设置成“*”即可。具体请参看如下代码中的 modify 方法。

```
public void modify(char[] chas) {
        if (chas == null || chas.length == 0) {
                return;
        }
        int j = chas.length - 1;
        for (int i = chas.length - 1; i > -1; i--) {
                if (chas[i] != '*') {
                        chas[j--] = chas[i];
                }
        }
        for (; j > -1;) {
                chas[j--] = '*';
        }
}
```

以上两道题目都是利用倒着复制这个技巧，其实很多字符串问题也和这个小技巧有关。字符串的面试题一般不会太难，很多题目都是考查代码实现能力的。

翻转字符串

【题目】

给定一个字符类型的数组 chas，请在单词间做逆序调整。只要做到单词顺序逆序即可，对空格的位置没有特别要求。

【举例】

如果把 chas 看作字符串为"dog loves pig"，调整成"pig Loves dog"。

如果把 chas 看作字符串为"I’m a student."，调整成"student. a I'm"。