算法exercise01

homework

1.

```
# -*- coding: UTF-8 -*-
#k为移动位数，n为数组长度，S为数组
def right_move01(k, n, S):
    p = 0 #附加的空间
    if k > n:
        k = k % n;
    for index in n:
        if index + k < n:
            p =  S[index + k];
            S[index + k] = S[index];
        else:
            p = S[(index + k) % n]
            S[(index + k) % n] = S[index]
#该算法时间复杂度是O(1)
```

2.

```
def right_move02(k, n, S):
    p = 0 #附加的空间
    if k > n:
        k = k % n;
    while k > 0:
        p = S[n-1]
        for i in range(n, -1, -1):
            S[i] = S[i-1]
        S[0] = p
        k -= 1
#该算法时间复杂度是O(k * n)~O(n)
```

3.

```
#逆序数组
def reverse(S, l, r):
    p = 0 #附加的空间
    while l == r:
        p = S[l]
        S[r] = S[l]
        S[l] = p
        l += 1
        r -= 1


def right_move03(k, n, S):
    if k > n:
        k = k % n;
    reverse(S, 0, n-k-1)
    reverse(S, n-k, n-1)
    reverse(S, 0, n-1)
#该算法时间复杂度是O(n)
```