日报格式:

主 题：日报

汇报人：杨明

日 期：2018年7月28日 星期六

内 容：

1. 今日学习了的内容：
2. 学习了shiro中的Authenticor以及AuthenticationStrategy验证规则
3. 日常一道牛客算法题。Java基础选择题大概四十道左右
4. Linux基础指令，rm、vim、输出重定向以及cat指令，还有df和free指令
5. 没有解决的问题：

1）AuthenticationStrategy身份验证里边的AtLeastOneSuccessfulStrategy和FirstSuccessfulStrategy的方法验证还没有通过。

1. 明天的学习计划：
2. 开始学习shiro里的授权
3. 继续学籍Linux里的进阶指令，多加练习
4. 牛客网算法题日常一道，加选择题若干

4.学习任务完成度： （95）%[注：0% ~ 100%]

5.对自己今天学习的满意度：（9）分 [注：1-10分]

6.对今天的学习做一句话的总结：

感觉今天是最充实的一天，因为感觉真的学到了东西

附录：

算法1：

题目： *\* 题目：输入一个链表，输出该链表中倒数第k个结点。*

{

*/\*\*  
 \* 两种方法如下  
 \*/*class ListNode {  
 int val;  
 ListNode next = null;  
  
 ListNode(int val) {  
 this.val = val;  
 }  
}  
public class Solution {  
 */\*\*  
 \* 运行时间：23ms  
 \* 占用内存：9696k  
 \*  
 \* 核心思路：  
 \* 先遍历一遍链表，记录下链表长度，计算倒数第K个结点的下标，再从head开始遍历到第k个结点，  
 \* 将引用指向第k个结点，将此引用返回即可  
 \** ***@return*** *\*/* public ListNode FindKthToTail(ListNode head, int k) {  
 int cnt = 0;  
 ListNode p = head;  
 //计算链表长度  
 while (p != null) {  
 ++cnt;  
 p = p.next;  
 }  
 if (k<1||k>cnt)//k不符合要求  
 return null;  
 p = head;//将p重新初始化为指向链表头  
 for (int i = 0;i<cnt && p!=null ;i++){  
 if (i == (cnt - k))//找到倒数第k个结点的下标  
 break;  
 p = p.next;//如果不是，p向后移动  
 }  
 return p;  
 }  
  
 */\*\*  
 \* 运行时间：22ms  
 \* 占用内存：9740k  
 \*  
 \* 核心思路：  
 \* 定义两个结点引用p、q，初始都指向head，先让一个引用q向后跑k-1个位置，此时p指向的是链表第k个结点，  
 \* 然后p和q同时向后跑，当q指向链表最后一个结点时，p的指向就是链表第k个结点，将其返回即可  
 \** ***@return*** *\*/* public ListNode FindKthToTail2(ListNode head, int k){  
 ListNode p = head;  
 ListNode q = p;  
 if (k<1||head==null)//k不符合要求或链表为空  
 return null;  
 for (int i=1;i<k;i++) { //先让q向后跑k-1个结点  
 if (q.next != null)  
 q = q.next;  
 else  
 return null; //这种情况为k超出链表最大长度的情况  
 }  
 //判断q的指向是否为最后一个结点，如果是，p和q同时后移  
 while (q.next!=null){  
 p = p.next;  
 q = q.next;  
 }  
 //循环结束，q指向链表最后一个结点，p指向倒数第k个结点  
 return p;  
 }  
  
  
 public static void main(String[] args){  
 ListNode head = new ListNode(0);  
 ListNode p = head;  
 for(int i=1;i<10;i++){  
 p.next = new ListNode(i);  
 p = p.next;  
 }  
 Solution solution = new Solution();  
// p = solution.FindKthToTail(head,1);  
 p = solution.FindKthToTail2(head,11);  
 System.*out*.println(p.val);  
 }  
}

【学习内容真实可靠，给自己的评分和完成度客观真实，不谦虚，不夸大】