1. 题目
2. 题干

给定一个链表，删除链表的倒数第 n 个节点，并且返回链表的头结点。

1. 示例

给定一个链表: **1->2->3->4->5**, 和 ***n* = 2**.

当删除了倒数第二个节点后，链表变为 **1->2->3->5**.

1. 题解
2. 思路

看到题目的一瞬间，删除链表倒数的第n个节点，就直接想起了双指针法。由于是要删除倒数第n个数，因此我们就要找到倒数第n+1个数，而不是倒数第n个数。首先让前面的指针向前走n+1个数，然后一直保持两个指针之间的距离，知道前面的指针指向了最后一个数，那么后面的指针一定指向了倒数第n+1个数，通过这个指针删除倒数第n个数即可。

1. 代码实现

Java :

import java.util.ArrayList;  
import java.util.Arrays;  
import java.util.List;  
  
public class fourNumSum18 {  
 public List<List<Integer>> fourSum(int[] nums, int target){  
 List<List<Integer>> quadruplets = new ArrayList<List<Integer>>();  
 if(nums==null || nums.length==0)  
 return quadruplets;  
 Arrays.*sort*(nums);  
 int length = nums.length;  
 for(int i = 0; i < length-3; ++i){  
 if(i > 0 && nums[i]==nums[i-1])  
 continue;  
 if(nums[i]+nums[i+1]+nums[i+2]+nums[i+3] > target)  
 break;  
 if(nums[i]+nums[length-3]+nums[length-2]+nums[length-1] < target)  
 continue;  
 for(int j = i+1; j <length-2; ++j){  
 if(j > i+1 && nums[j]==nums[j-1])  
 continue;  
 if(nums[i]+nums[j]+nums[j+1]+nums[j+2] > target)  
 break;  
 if(nums[i]+nums[j]+nums[length-2]+nums[length-1] < target)  
 continue;  
 int left = j+1, right=length-1;  
 while(left < right){  
 int sum = nums[i]+nums[j]+nums[left]+nums[right];  
 if(sum == target){  
 quadruplets.add(Arrays.*asList*(nums[i],nums[j],nums[left],nums[right]));  
 while(left <right && nums[left]==nums[left+1])  
 left++;  
 left++;  
 while(left < right && nums[right]==nums[right-1])  
 right--;  
 right--;  
 }  
 else if(sum < target)  
 left++;  
 else  
 right--;  
 }  
 }  
 }  
 return quadruplets;  
 }  
}