1、顺序表和单链表的就地逆置算法流程：

对于顺序表，可以通过交换首尾元素、次首次尾元素，以及依次类推的方式进行逆置。使用两个指针left和right，初始时left指向表头，right指向表尾，然后不断交换left和right指向的元素，并将left指针右移一位，right指针左移一位，直到left指针超过或等于right指针。

对于单链表，可以通过修改每个节点的指针来实现逆置。使用三个指针prev、curr和next，初始时prev指向null，curr指向表头，next指向curr的下一个节点，然后不断将curr的next指针指向prev，然后依次将prev、curr和next指针右移一位，直到curr指针指向null。

Procedure ReverseList(List) // 逆置顺序表

left := 0

right := length(List) - 1

while left < right do

swap(List[left], List[right])

left := left + 1

right := right - 1

end while

End Procedure

Procedure ReverseLinkedList(LinkedList) // 逆置单链表

prev := null

curr := head(LinkedList)

while curr != null do

next := next(curr)

next(curr) := prev

prev := curr

curr := next

end while

head(LinkedList) := prev

End Procedure

2、小球移动游戏伪代码如下所示：

Procedure MoveBalls(A, B, C, n)

// A为起始桶，B为目标桶，C为中转桶，n为球的数量

if n == 1 then

// 当只有一个球时，直接从A放到B

PutBall(A, B)

else

MoveBalls(A, C, B, n-1) // 将前n-1个球从A放到C，借助B作为中转桶

PutBall(A, B) // 将第n个球从A放到B

MoveBalls(C, B, A, n-1) // 将C中的球借助A作为中转桶，放到B中

end if

End Procedure

3、铁钉排序存储结构和算法思路：

可以使用数组作为数据结构来存储这些钉子，将每个钉子的长度作为数组的元素值。然后使用常见的排序算法（如冒泡排序、插入排序等）对数组进行升序排序。

算法思路：

遍历数组，比较相邻的两个元素大小，如果顺序不正确，则交换位置。

继续遍历，直到整个数组排序完成。

伪代码如下所示：

Procedure SortNails(Nails, K)

for i := 1 to N-1 do

for j := 0 to N-i-1 do

if Nails[j] > Nails[j+1] then

Swap(Nails[j], Nails[j+1])

end if

end for

end for

End Procedure

算法的时间复杂度为O(N^2)，空间复杂度为O(1)。