Assignment 1 Paul (Pengshengnan) Cheng V00838497

Question 1

a)

Algorithm reverse()
    newhead ← reverse1(head)
    head.next ← null
    head ← newhead

Algorithm reverse1(node)
① If node.next ≠ null then
②   newhead ← reverse1(node.next)
③   node.next.next ← node
④   return newhead
  else
⑤   return node
  end if

b) For reverse()
① 1+1 assignment + function call
② 1+1 assignment + .next
③ 1 assignment

For reverse1() run (n-1) times
① Loop: 1+1 .next + comparison
② end loop 1+1+1 assignment + function call + .next
③ 1+1+1 assignment + 2 .next
④ 1 return
⑤ 1 return

$T_1(n) = T_2(n) + 5$

$T_2(n) = 3 \quad (n=1)$
$= T_2(n-1) + 9 \quad (n=2)$

c) $T_2(n) = T_2(n-1) + 9 \quad n \geq 2$

$T_2(2) = T_2(1) + 9$

$T_2(3) = T_2(2) + 9 = T_2(1) + 9 + 9$

$T_2(4) = T_2(3) + 9 = T_2(2) + 9 + 9 = T_2(1) + 9 + 9 + 9$

$\vdots$

$T_2(n) = T_2(1) + (n-1) \cdot 9$

$= 3 + 9n - 9$

$= 9n - 6$

$T_1(n) = T_2(n) + 5$

$= 9n - 6 + 5$

$= 9n - 1$

$T_1(n) = 9n - 1 = O(n)$

Question 2

a)

```
reverse( )
If head = null then
return null
end if
If head.next = null then
return head
end if
prev ← head
curr ← prev.next
next ← curr.next
prev.next ← null
curr.next ← prev
```

```
while    next ≠ null do
        prev ← curr
        curr ← next
        curr.next ← prev
        next ← next.next
end while
head ← curr
end
```

b) In general, we should study the function and understand it;
Then try convert recursive calls into tail calls;
Try to use while or for loops instead recursive calls;
Convert tail calls into continue statement;
Then try to clean up the function.

Question 3

array contains $n-1$ unique integers in $[0, n-1]$
So if we add back the missing integer, we will get a arithmetic sequence $a_1 = 0$ $d = 1$
Sum of 0 to $n-1$ is $\frac{n(0+n-1)}{2}$

```
Algorithm Findmiss (A)
    array_sum ← 0
    Sum ← n(n-1)/2
    from i=0 to i=n-1 do
    array_sum ← array_sum + A[i]
    return sum − array_sum
```

Question 4

a) $T(n) = 4 \quad n \leq 2$ n is Length of the Array

$= 2+3+2+T(\frac{n}{2})+3+T(\frac{n}{2})+2$

$= 2T(\frac{n}{2})+12 \quad n>2$

b) $T(n) = 2T(\frac{n}{2})+12$

$a=2 \quad b=2 \quad c=12 \quad d=0$

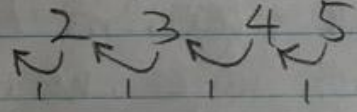
c) $a=2 \quad b^d = 1$

$a > b^d$

So $T(n) = O(n^1) = O(n)$

Question 5

a) insertion sort



4 times

Merge sort

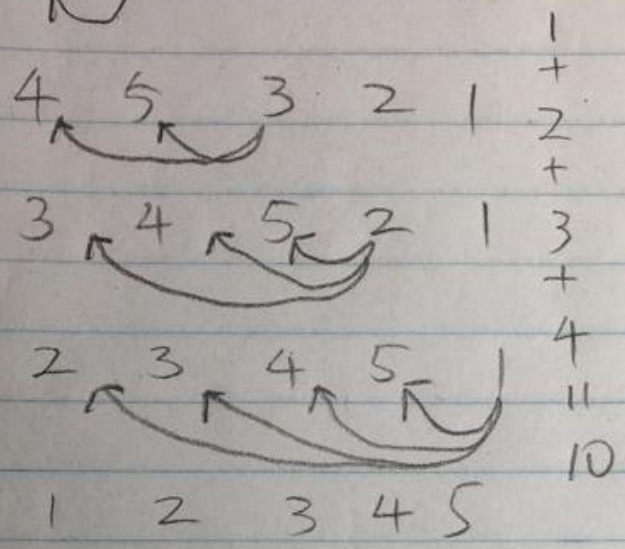
$\{1,2\}, \{3,4\}, \{5\}$ 2

$\{1,2,3,4\} \{5\}$ 2

$\{1,2,3,4,5\}$ 4

2+2+4 = 8 times

b) 5 4 3 2 1



1 2 3 4 5

10 times

c) Worst-case for insertion sort is Reverse
Suppose we have n elements in Reverse
it will compare $0+1+2+3 \ldots +(n-1)$ times.
it's a Arithmetic sequence

$$\text{sum} = \frac{[(n-1)+0]\cdot n}{2} = \frac{n\cdot(n-1)}{2} = \frac{n^2-n}{2}$$

$$\text{sum} = O(n^2)$$