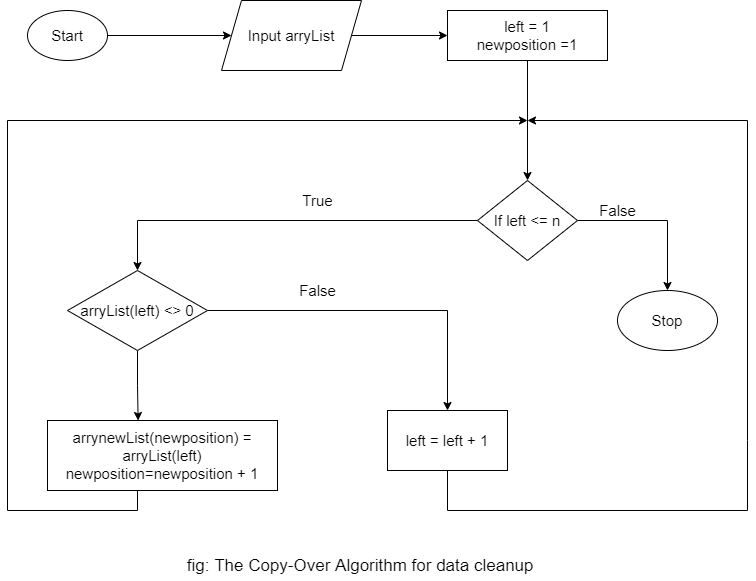
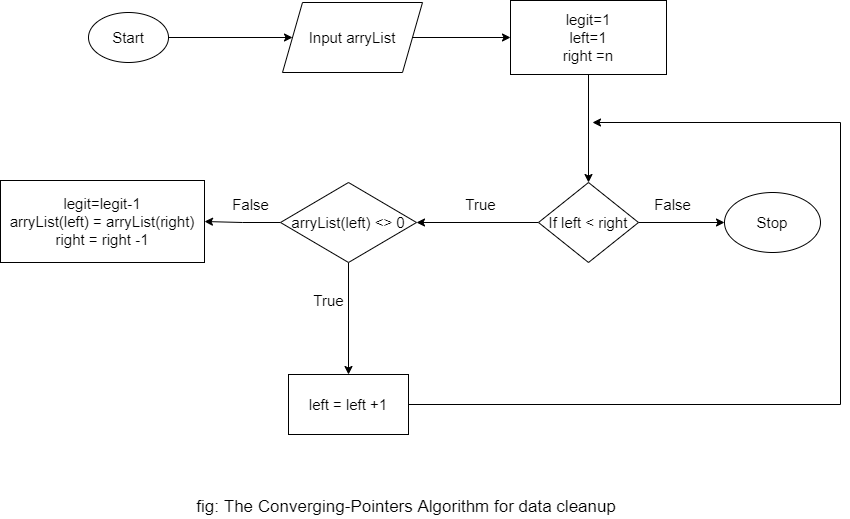
Q.1.

a.



b)



2)

Big O for Insertion sort algorithm:

1. Worst Case Performance: O (n^2)
2. Best Case Performance: O (n)
3. Average Case Performance: O (n^2)
4. Worst Case Space Complexity: O (1)

3)

Elements in the Array= {0, 9, 88, 7, 0, 5, 3, 19, 7}.

To Do:  **Insertion sort in ascending order**.

Step 1:

Divide the array into a left, sorted part, and a right, unsorted part.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

Left (Sorted) Right (Unsorted)

Step 2:

Look at primary element of the unsorted area and check where, within the sorted area, it must be inserted by comparing it with its left neighbor. So, inserting 9 in the sub list.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

Step 3:

9>0 So, no change in the position. Sorted list (0,9). Inserting 88 now

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

Step 4:

Sorted list= {0,9,88}. Inserting 7 which is > 0 but < 9 and 88. Interchange the position.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

Step 5:

Sorted list= {0,7,9,88}. Inserting 0 which is < 7, 9 and 88. Interchanging the positions

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 7 | 9 | 88 | 0 | 5 | 3 | 19 | 7 |

Step 6:

Sorted list={0,0,7,9,88}. Inserting element is 5 which is > 0 and < 7,9 and 88. Thus we move 5 to the position of 7.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 7 | 9 | 88 | 5 | 3 | 19 | 7 |

Step 6:

The sorted sub list is now {0,0,5,7,9,88}. Inserting element 3 between 0 and 5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 5 | 7 | 9 | 88 | 3 | 19 | 7 |

Step 7:

Sorted list= {0,0,3,5,7,9,88}. Inserting 19 which is < 88 and > 9.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 3 | 5 | 7 | 9 | 88 | 19 | 7 |

Step 8:

List: {0,0,3,5,7,9,19,88}. Lastly, 7 which is >5 but <9.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 3 | 5 | 7 | 9 | 19 | 88 | 7 |

Step 9:

The sorted sublist is {0,0,3,5,7,7,9,19,88}. Now our array is completely sorted and the sorted array is:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 3 | 5 | 7 | 7 | 9 | 19 | 88 |

4)

Answer:

The given array is {0,9,88,7,0,5,3,19,7}. Since the list contains 0 value, which is called non legitimate data and we remove such data-by-data cleanup process.

**Shuttle Left Algorithm**

Step 1:

In shuffle left algorithm, the list is preceded from left to right with a left point and a right point. We proceed through the list from left to right, by pointing a finger on the left and passing it over nonzero values. After a 0 value is encountered, we remove it out of the list by copying each remaining data item in the list one cell to the left. And

The given array is:

Legit = 9

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

First cell contains 0, so legit= legit- 1, and all of the items to the right of the 0 must be copied one cell left. Hence after copy

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

2nd copy

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

3rd copy

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 7 | 0 | 5 | 3 | 19 | 7 |

Last item copy

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 | 7 |

Resetting as nothing is left to copy:

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 | 7 |

Moving along, passing over the 88

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 | 7 |

Cycle of 5 copies takes place to squeeze out the 0; the result is

Legit = 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 | 19 | 7 | 7 | 7 |

Result

Legit = 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 | 19 | 7 | 7 | 7 |

Shuttle left Algorithm where legit =7.

5)

Answer:

Array= {0,9,88,7,0,5,3,19,7}.

**COPY OVER ALGORITHM**

Step 1:

The first entry list is 0 so we omit is and place the value of 2nd array list i.e., 9

|  |
| --- |
| 9 |

Step 2:

If array list[value]! = 0 then put it in the list,

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 |  |  |  |  |  |  |  |  |

Step 3:

Similarly, as step 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 | 19 |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 | 19 | 7 |  |  |  |

Result of copy over algorithm

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 88 | 7 | 5 | 3 | 19 |  |  |  |  |

6)

Array={0,9,88,7,0,5,3,19,7}.

**Convergent-pointer Algorithm**

Process:

1) assign Left and Right and place them into first and last values into list.

2) The left finger slides to the right over non zero values.

3) Whenever the left finger encounters a 0 item,

Legit = legit- 1

Copy from right finger into left finger position and slide right finger one cell left

Legit=9

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

(L) (R)

0 is encountered. So, copy 7 in place of 0. Move position of R to 19 and L to 9. Legit=legit-1.

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

(L) (R)

We can’t find 0. Position(L) is increased.

Legit = 8

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 9 | 88 | 7 | 0 | 5 | 3 | 19 | 7 |

(L) (R)

Replace 0 by 19 and shift the position or R and L. Reduce legit by 1.

Legit = 7

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 9 | 88 | 7 | 19 | 5 | 3 | 19 | 7 |

(L) (R)

Since, Position of L and R are same, Algorithm is complete.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 9 | 88 | 7 | 19 | 5 | 3 | 19 | 7 |

(L) (R)

7)

Array= {3,0,0,2,6,7,0,0,5,1}

**Shuttle Left Algorithm**

Legit=10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 0 | 0 | 2 | 6 | 7 | 0 | 0 | 5 | 1 |

Finds 0 Finds 0 Finds 0 Finds 0

P2 P3 P7 P8

Shift values from right to left where there is 0. legit is reduced by 1, every time 0 is encountered.

Steps:

Legit = 10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 0 | 0 | 2 | 6 | 7 | 0 | 0 | 5 | 1 |

Legit = 9,

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 0 | 2 | 6 | 7 | 0 | 0 | 5 | 1 | 1 |

Legit = 8

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 2 | 6 | 7 | 0 | 0 | 5 | 1 | 1 | 1 |

Legit = 7

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 2 | 6 | 7 | 0 | 5 | 1 | 1 | 1 | 1 |

Legit = 6

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 2 | 6 | 7 | 5 | 1 | 1 | 1 | 1 | 1 |

Since, all 0 have been replaced and filled by right cell i.e., 1. This is our required shuffle left algorithm.

Legit is reduced after 0 has been removed.

Therefore, total items = 8+7+3+2+4=26