Practice Assignment – 3 Algorithms and Problem-Solving

Task 1:

BEGIN

STEP 1. Set N = Length of Array

STEP 2. if Array[j] > Array[j + 1]

    if no - continue with the step 4

STEP 3. Swap Array[j] and Array[j+1]

STEP 4. Increment j (j++)

STEP 5. if j < N - i - 1

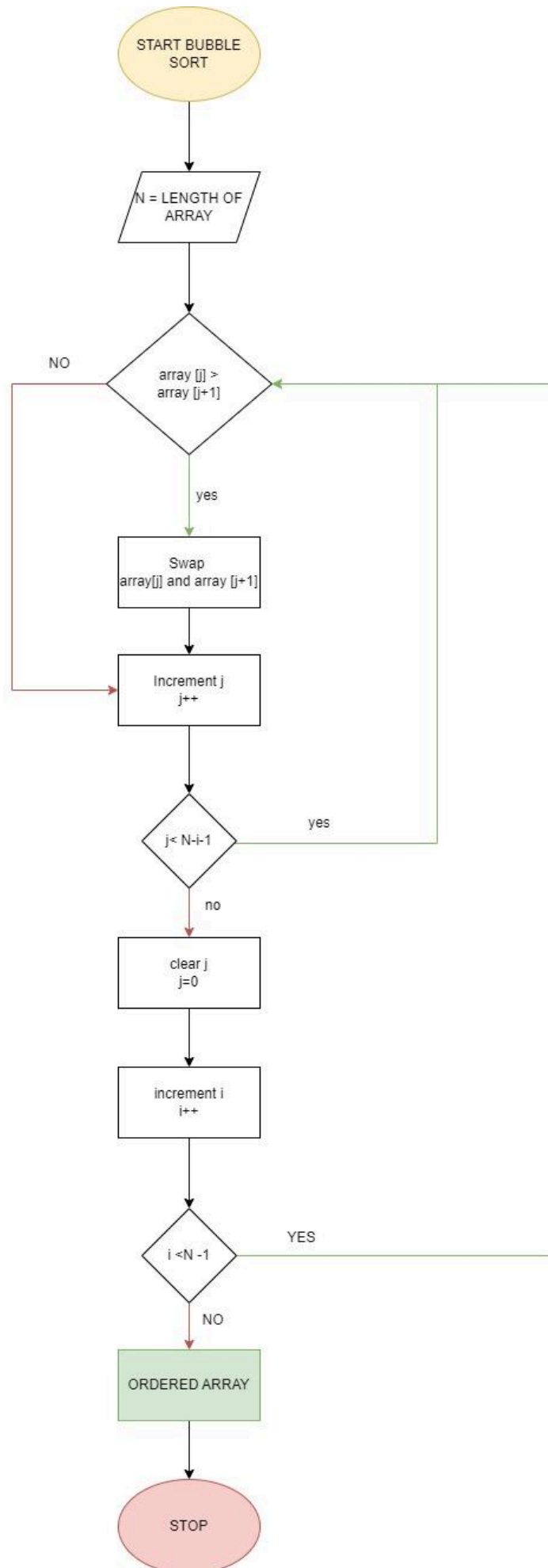   if yes - continue with the step 2

STEP 6. Clear j (j = 0)

STEP 7. Increment i (i++)

STEP 8. If i < N - 1

    if yes continue with step 2

STEP 9. OUTPUT: Ordered Array

END

```
                    START BUBBLE
                       SORT
                         │
                         ▼
                  N = LENGTH OF
                     ARRAY
                         │
                         ▼
    NO ◀──────────  array [j] >
                    array [j+1]  ◀─────────────────┐
                         │                          │
                        yes                         │
                         ▼                          │
                       Swap                         │
                 array[j] and array [j+1]           │
                         │                          │
                         ▼                          │
                   Increment j                      │
    └─────────────▶     j++                         │
                         │                          │
                         ▼                          │
                     j< N-i-1  ─────── yes ─────────┘
                         │
                        no
                         ▼
                      clear j
                       j=0
                         │
                         ▼
                   increment i
                       i++
                         │
                         ▼
                     i <N -1  ─────── YES ──────────┐
                         │
                        NO
                         ▼
                  ORDERED ARRAY
                         │
                         ▼
                       STOP
```

DRY RUN

INPUT: 13,32,26,35,10

STEP 1. 13>32? GIVES: No   =>13,32,26,35,10
STEP 2. 32> 26? GIVES: Yes => SWAP : 13,26,32,35,10
STEP 3. 32>35? GIVES: No   =>13,26,32,35,10
STEP 4. 35> 10? GIVES: Yes => SWAP : 13,26,32,10,35

2nd iteration
STEP 1. 13> 26? GIVES: No   =>13,26,32,10,35
STEP 2. 26>32? GIVES: No   =>13,26,32,10,35
STEP 3. 32>10? GIVES: YES   =>SWAP : 13,26,10,32,35

3rd iteration
STEP 1. 13>26? GIVES: No   =>13,26,10,32,35
STEP 2. 26>10? GIVES: YES   =>SWAP : 13,10,26,32,35

4th iteration
STEP 1. 13?10? GIVES: YES   =>SWAP : 10,13,26,32,35

EXPLANATION :

BUBBLE SORT is an algorithm to sort a given set of arrays. When any input is given it repeatedly swaps the adjacent elements until they are not in the intended order.
In the dry run, we took the array [13,32,26,35,10]. First 13 and 32 are compared, as 13 is less than 32, so no swapping is done. Then 32 and 26 are compared, as 32 is larger than 26 then swapping is done. Iteration is done till a fully ordered array is achieved.

Task 2:

Recursive function

Fibo(n)
Begin

  if n <= 1 then
      Return n;
  else
      Return Call Fibo(n-1) + Call Fibo(n-2);
  Endif

End

The time complexity is O(2^n)
The Space Complexity is O(n)

 For iterative approach

BEGIN

int fib(int n)
   int a = 1, b = 1;
   for (int i = 3; i <= n; i++) {
      int c = a + b;
      a = b;
      b = c;
   return b;

END

The time complexity is O(n)
The Space Complexity is O(1)

The use of the iterative method is better for time and space-optimized