**Bubble Sort Implementation in C**

**Objective:**

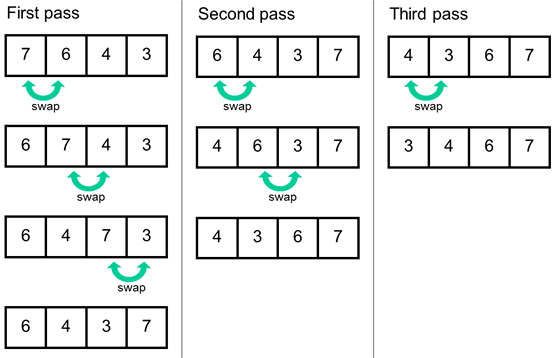
To implement the Bubble Sort algorithm in C and analyze its time complexity.

**Theory:**

Bubble Sort is a simple sorting algorithm that repeatedly iterates through a list, compares adjacent elements, and swaps them if they are in the wrong order. This process continues until the list is sorted.

**Algorithm:**

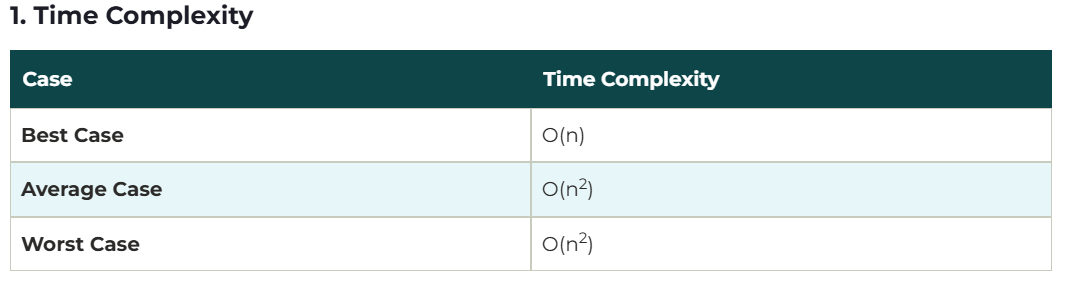
1. begin BubbleSort(arr)
2. **for** all array elements
3. **if** arr[i] > arr[i+1]
4. swap(arr[i], arr[i+1])
5. end **if**
6. end **for**
7. **return** arr
8. end BubbleSort

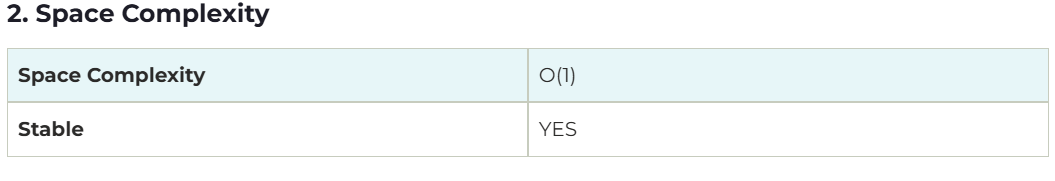


**Implementation**

1. #include<stdio.h>
2. **void** print(**int** a[], **int** n) //function to print array elements
3. {
4. **int** i;
5. **for**(i = 0; i < n; i++)
6. {
7. printf("%d ",a[i]);
8. }
9. }
10. **void** bubble(**int** a[], **int** n) // function to implement bubble sort
11. {
12. **int** i, j, temp;
13. **for**(i = 0; i < n; i++)
14. {
15. **for**(j = i+1; j < n; j++)
16. {
17. **if**(a[j] < a[i])
18. {
19. temp = a[i];
20. a[i] = a[j];
21. a[j] = temp;
22. }
23. }
24. }
25. }
26. **void** main ()
27. {
28. **int** i, j,temp;
29. **int** a[5] = { 10, 35, 32, 13, 26};
30. **int** n = **sizeof**(a)/**sizeof**(a[0]);
31. printf("Before sorting array elements are - \n");
32. print(a, n);
33. bubble(a, n);
34. printf("\nAfter sorting array elements are - \n");
35. print(a, n);
36. }

**Analysis:**



****

**Conclusion**

Bubble Sort is a simple sorting algorithm, but it's not very efficient for large datasets due to its quadratic time complexity in the average and worst cases. It's better suited for small datasets or as a teaching tool to understand basic sorting concepts.