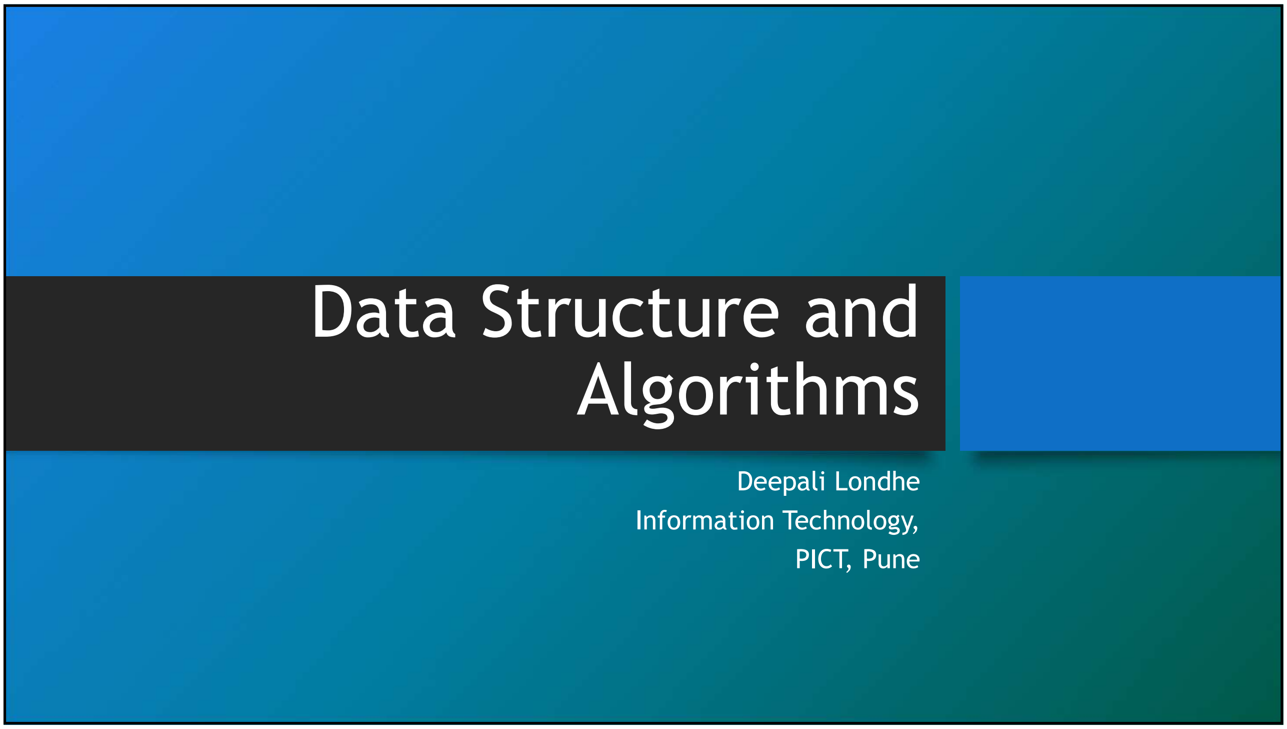
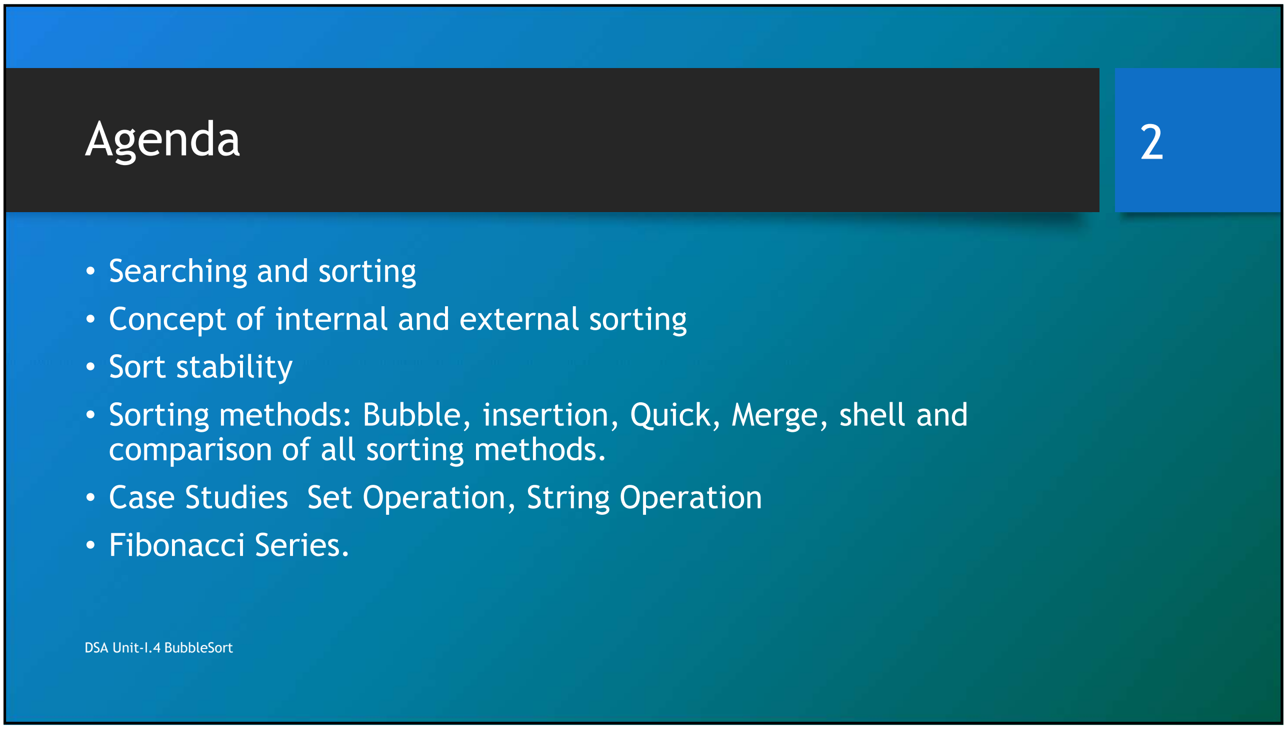


Data Structure and Algorithms

Deepali Londhe
Information Technology,
PICT, Pune

1



Agenda

2

- Searching and sorting
- Concept of internal and external sorting
- Sort stability
- Sorting methods: Bubble, insertion, Quick, Merge, shell and comparison of all sorting methods.
- Case Studies Set Operation, String Operation
- Fibonacci Series.

DSA Unit-I.4 BubbleSort

2

Bubble sort

3

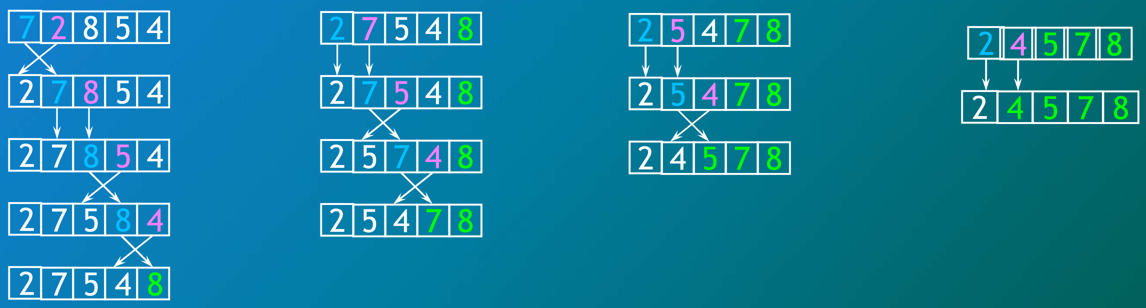
- Compare each element (except the last one) with its neighbor to the right
  - If they are out of order, swap them
  - This puts the largest element at the very end
  - The last element is now in the correct and final place
- Compare each element (except the last *two*) with its neighbor to the right
  - If they are out of order, swap them
  - This puts the second largest element next to last
  - The last two elements are now in their correct and final places
- Compare each element (except the last *three*) with its neighbor to the right
  - Continue as above until you have no unsorted elements on the left

DSA Unit-I.4 BubbleSort

3

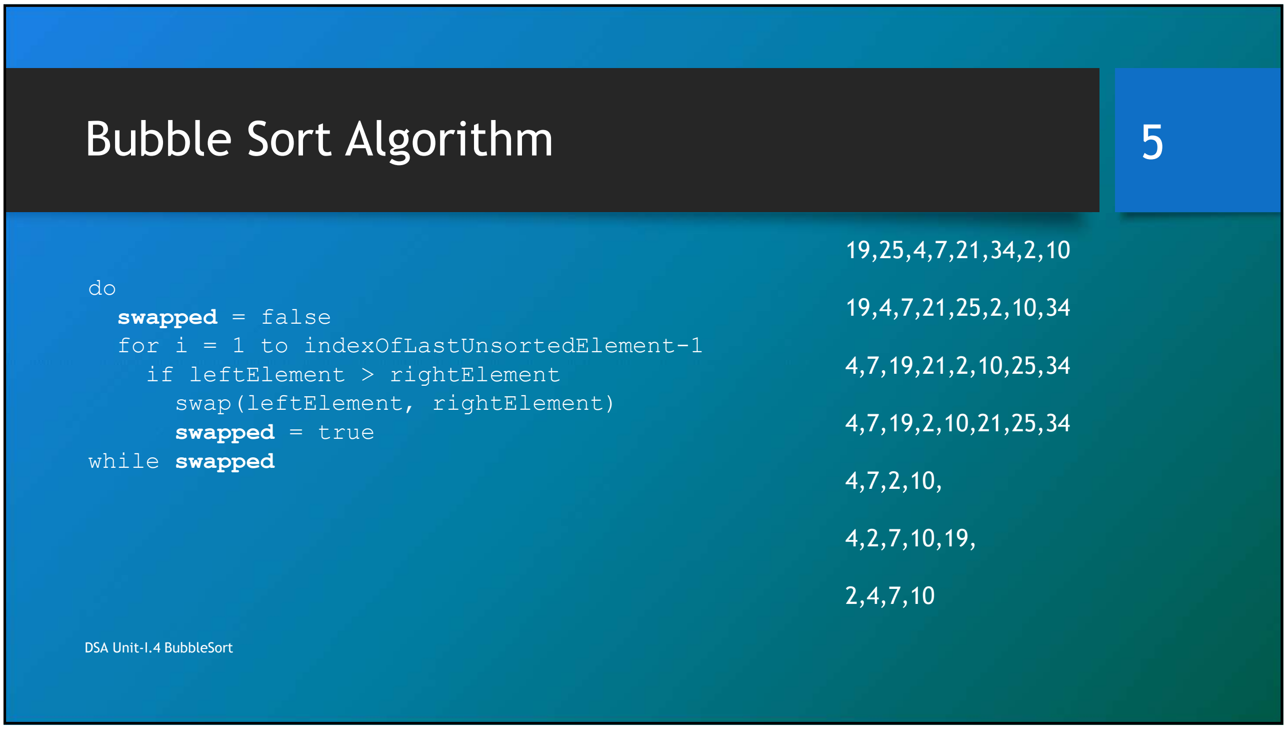
Bubble sort(Example)

4



DSA Unit-I.4 BubbleSort

4



Bubble Sort Algorithm

5

```
do
  swapped = false
  for i = 1 to indexOfLastUnsortedElement-1
    if leftElement > rightElement
      swap(leftElement, rightElement)
      swapped = true
while swapped
```

19,25,4,7,21,34,2,10

19,4,7,21,25,2,10,34

4,7,19,21,2,10,25,34

4,7,19,2,10,21,25,34

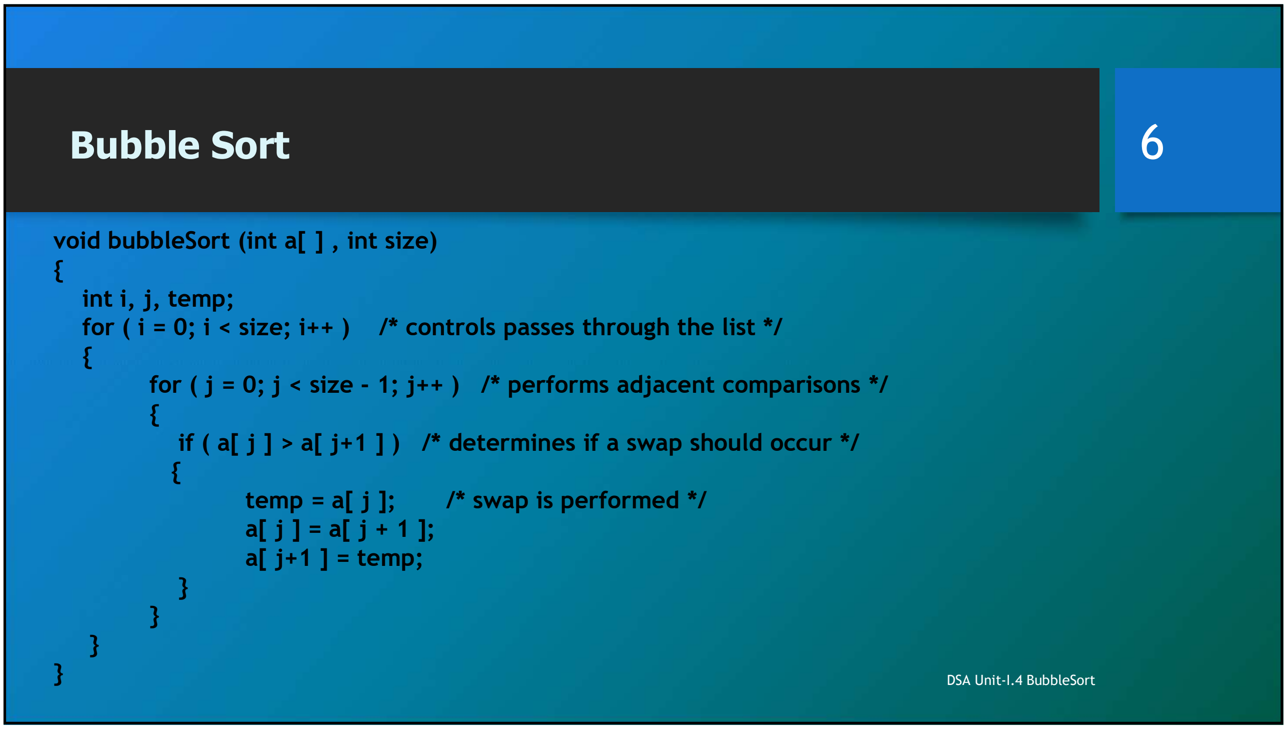
4,7,2,10,

4,2,7,10,19,

2,4,7,10

DSA Unit-I.4 BubbleSort

5



Bubble Sort

6

```
void bubbleSort (int a[ ] , int size)
{
   int i, j, temp;
   for ( i = 0; i < size; i++ )    /* controls passes through the list */
   {
          for ( j = 0; j < size - 1; j++ )   /* performs adjacent comparisons */
          {
             if ( a[ j ] > a[ j+1 ] )   /* determines if a swap should occur */
            {
                    temp = a[ j ];       /* swap is performed */
                    a[ j ] = a[ j + 1 ];
                    a[ j+1 ] = temp;
             }
          }
    }
}
```

DSA Unit-I.4 BubbleSort

6

Bubble Sort Passes

7

a[]={5,4,3,2,1}

pass 0	4	3	2	1	5
pass 1	3	2	1	4	5
pass 2	2	1	3	4	5
pass 3	1	2	3	4	5
sorted array	1	2	3	4	5

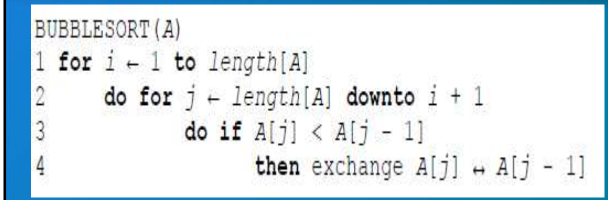
a[]={1,4,3,2,5}

pass 0	1	3	2	4	5
pass 1	1	2	3	4	5
pass 2	1	2	3	4	5
pass 3	1	2	3	4	5
sorted array	1	2	3	4	5

7

Bubble Sort Analysis

8



```
BUBBLESORT(A)
for i ← 1 to length[A]
    do for j ← length[A] downto i + 1
           do if A[j] < A[j - 1]
                  then exchange A[j] ↔ A[j - 1]
```

Outer loop- n

Inner Loop- (n-1), (n-2)…..1

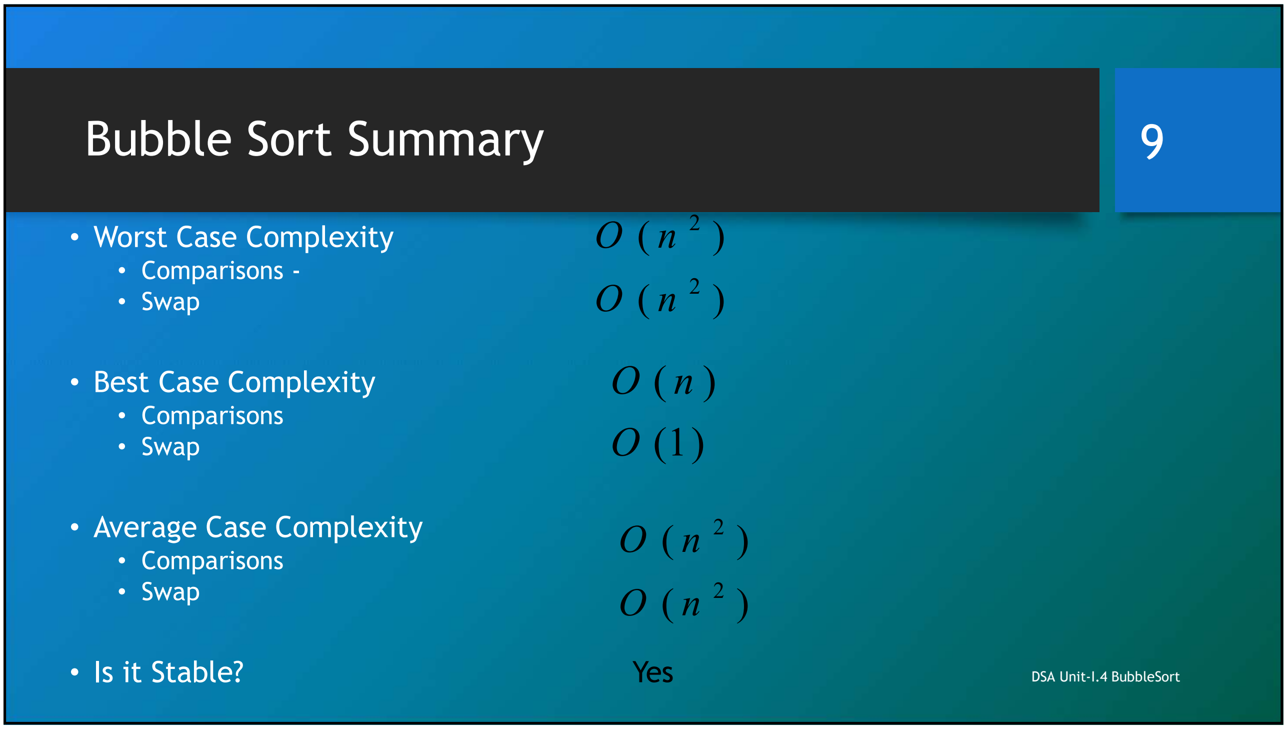
Comparison- Constant Time
Swap- Constant Time

- The outer loop is executed n-1 times (call it n, that's close enough)
- Each time the outer loop is executed, the inner loop is executed
  - Inner loop executes n-1 times at first, linearly dropping to just once
  - On average, inner loop executes about n/2 times for each execution of the outer loop
  - In the inner loop, the comparison is always done (constant time), the swap might be done (also constant time)

$$n * (n / 2) * k$$

$$O(n^2 / 2 * k) \sim O(n^2)$$

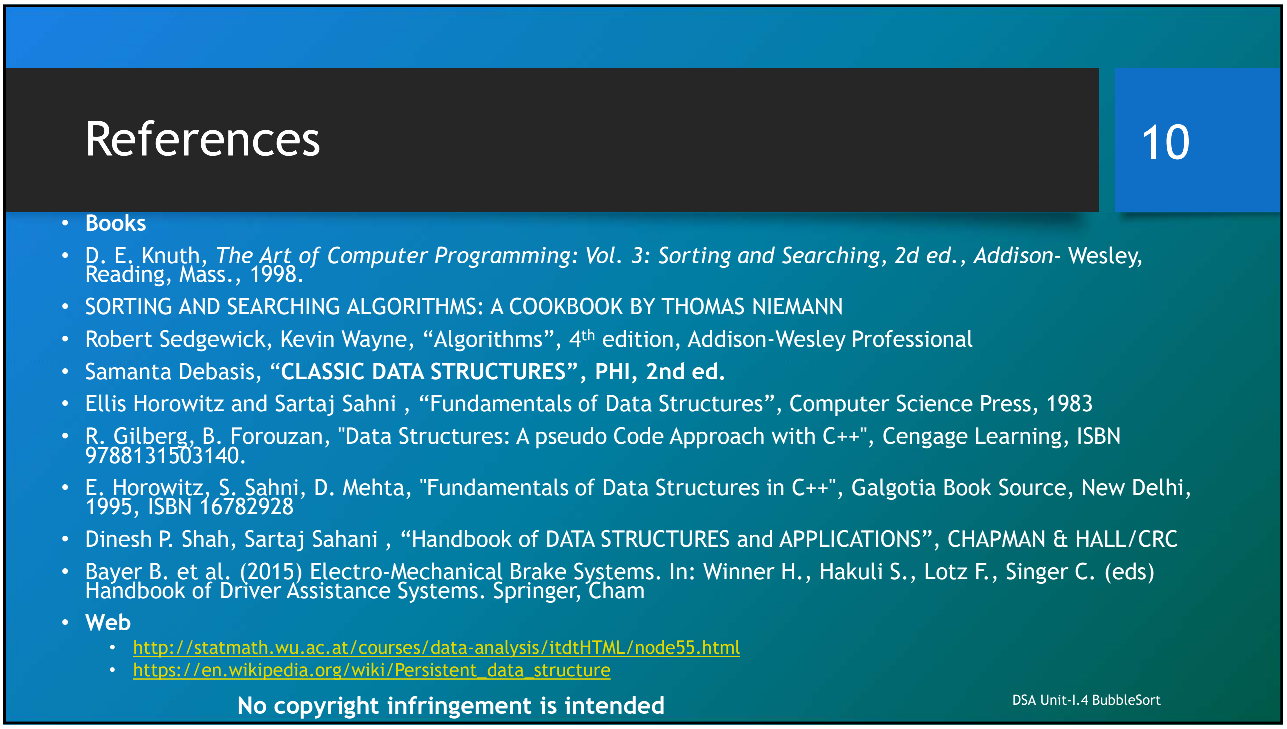
8



Bubble Sort Summary

9

• Worst Case Complexity	$O(n^2)$
• Comparisons -	
• Swap	$O(n^2)$
• Best Case Complexity	$O(n)$
• Comparisons	$O(1)$
• Swap	
• Average Case Complexity	$O(n^2)$
• Comparisons	
• Swap	$O(n^2)$
• Is it Stable?	Yes

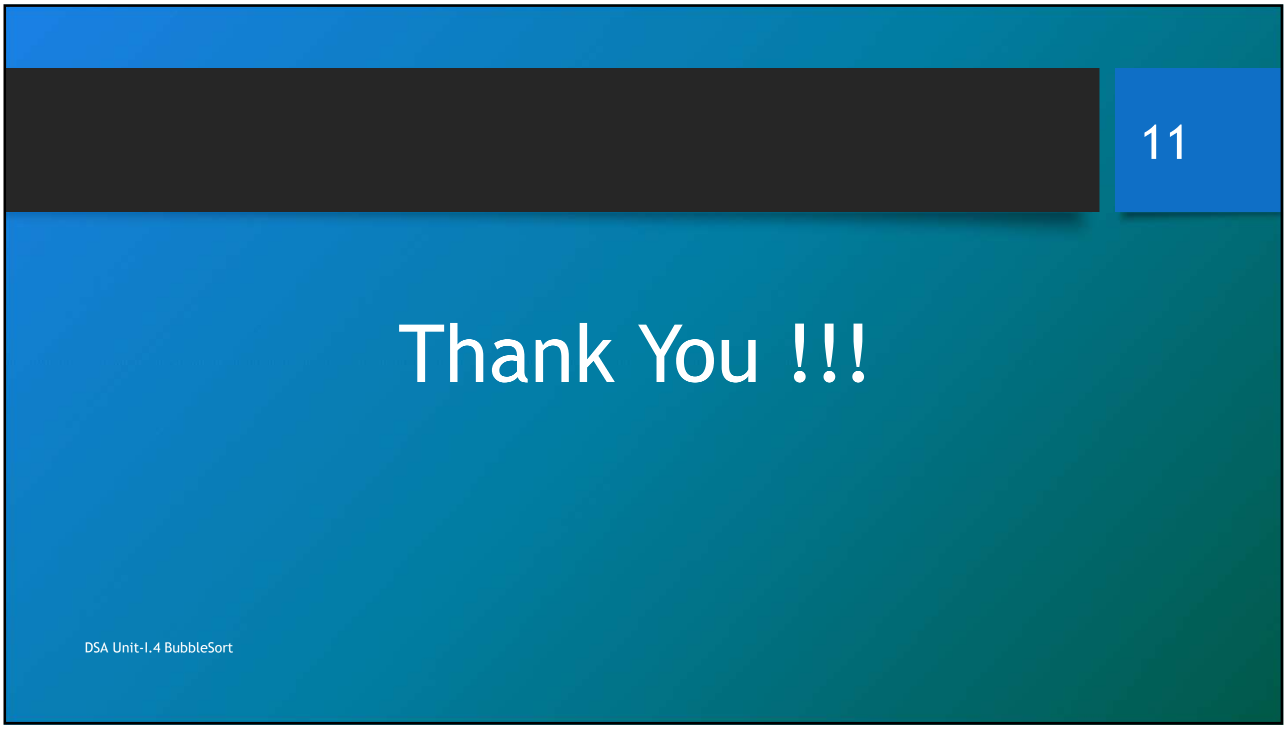


References

10

- **Books**
- D. E. Knuth, *The Art of Computer Programming: Vol. 3: Sorting and Searching, 2d ed., Addison-* Wesley, Reading, Mass., 1998.
- SORTING AND SEARCHING ALGORITHMS: A COOKBOOK BY THOMAS NIEMANN
- Robert Sedgewick, Kevin Wayne, “Algorithms”, 4th edition, Addison-Wesley Professional
- Samanta Debasis, “**CLASSIC DATA STRUCTURES**”, **PHI, 2nd ed.**
- Ellis Horowitz and Sartaj Sahni , “Fundamentals of Data Structures”, Computer Science Press, 1983
- R. Gilberg, B. Forouzan, "Data Structures: A pseudo Code Approach with C++", Cengage Learning, ISBN 9788131503140.
- E. Horowitz, S. Sahni, D. Mehta, "Fundamentals of Data Structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN 16782928
- Dinesh P. Shah, Sartaj Sahani , “Handbook of DATA STRUCTURES and APPLICATIONS”, CHAPMAN & HALL/CRC
- Bayer B. et al. (2015) Electro-Mechanical Brake Systems. In: Winner H., Hakuli S., Lotz F., Singer C. (eds) Handbook of Driver Assistance Systems. Springer, Cham
- **Web**
  - http://statmath.wu.ac.at/courses/data-analysis/itdtHTML/node55.html
  - https://en.wikipedia.org/wiki/Persistent_data_structure

No copyright infringement is intended



11

Thank You !!!

DSA Unit-I.4 BubbleSort

11