DAA - LAB

1. Sorting Algorithms (Divide and Conquer)
2. Greedy Algorithms.
3. Dynamic Programming based Algorithm.
4. Graph Algorithm
5. String matching Algorithms.

Sorting

- Comparision Based
  1. Selection sort
  2. Bubble sort
  3. Insertion sort
  4. Shell sort
  5. Heap sort
  6. Quick sort
  7. Merge sort
- Non comparision based
  1. Bucket sorting
  2. Redix sorting
  3. Counting sort

---

1. Selection sort

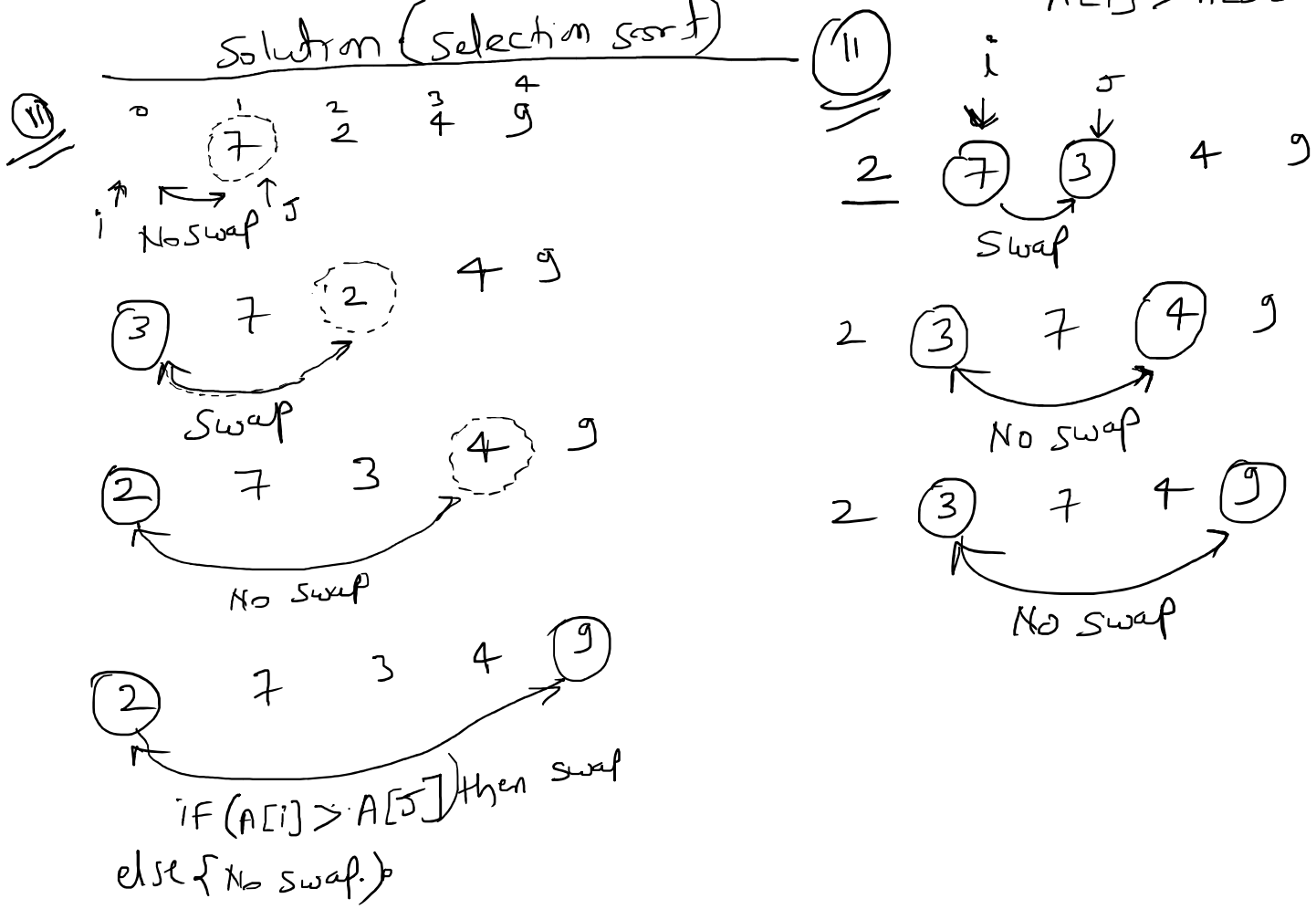
Problem definition: A[5] | 3, 7, 2, 4, 9 |

sort

A[5] | 2, 3, 4, 7, 9 |

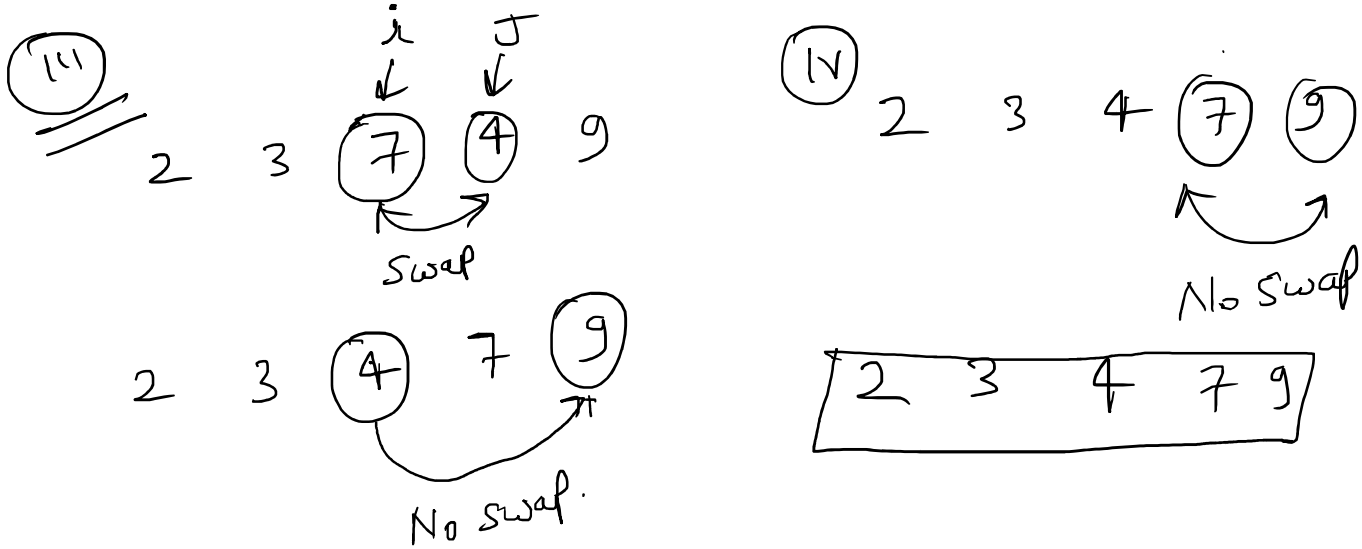
A[i] > A[j]

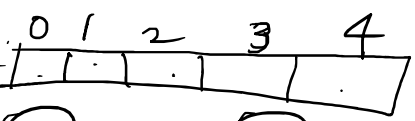
Solution (selection sort)



IF (A[i] > A[j]) then swap

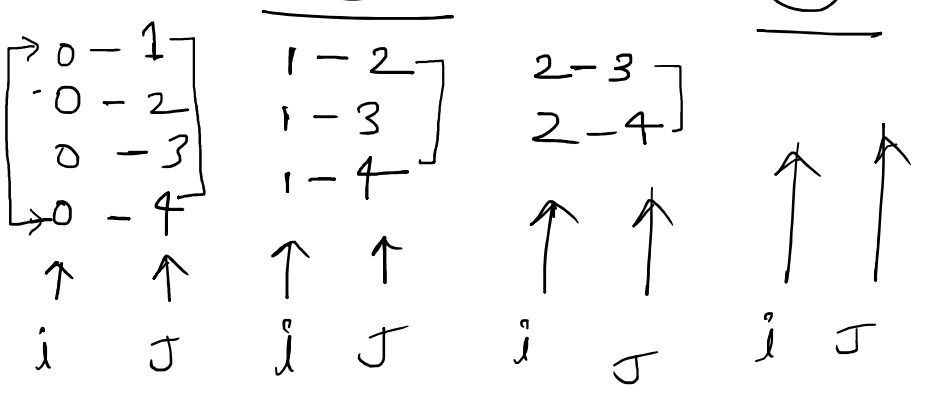
else { No swap. }

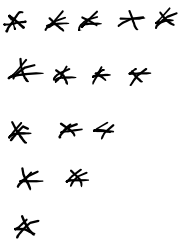


A[5] = 

Comparisons: (I) (II) (III) (IV)

Comparisons → (I) (II) (III) (IV)





(i+1)

IF(A[i] > A[J]) { Swap (A[i], A[J]) }

else { No swap }

n=5, n-1=4

for (i=0; i < n-1 ; i++) // i=0,1,2,3,4=5 // n

{

for (J = i+1 ; J < n ; J++) // J=1,2,3,4={4} // J=2,3,4={3} // J=3,4={2} // J=4={1}

{ if (A[i] > A[J]) //

{ temp = A[J] //

A[J] = A[i] //

A[i] = temp //

}

}

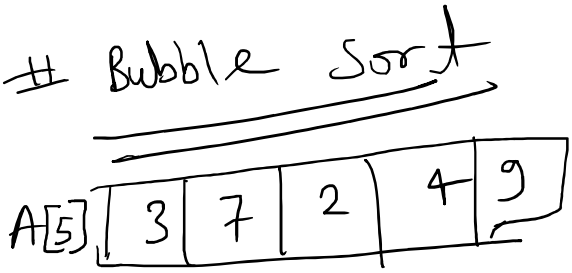
}

$$T(n) = n \cdot c_1 + \sum_{i=0}^{n-1} (n-i) * 4(n-1)$$

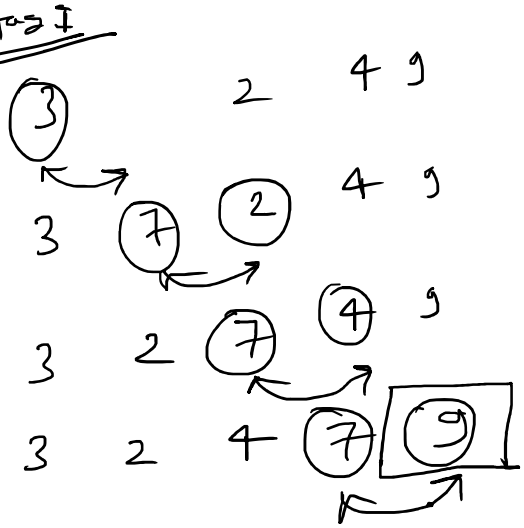
$$= n \cdot c_1 + n^2(\text{----})$$

$$= \Theta(n^2)$$

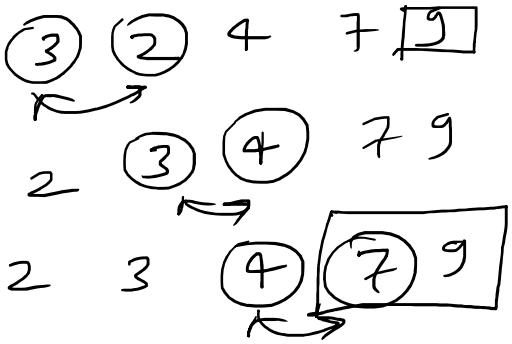
Bubble Sort



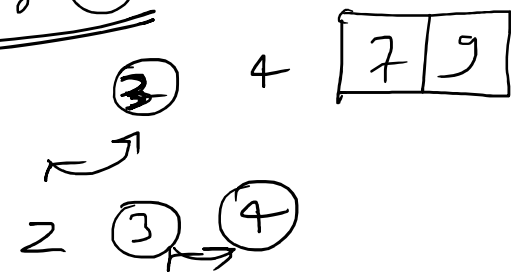
Stage I



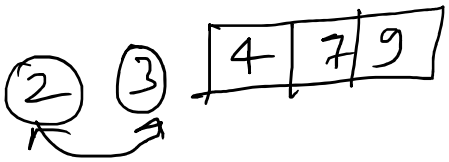
Stage II

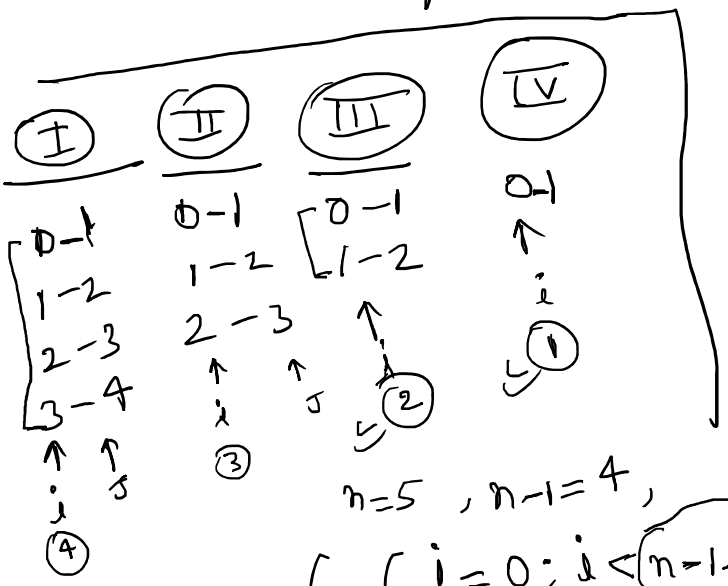


Stage III



Stage IV






```
n=5 , n-1=4,
for(i=0; i<(n-1-i); i++)
{ For (J=i+1; J<(n-i) ; J++)
  { if (A[i] > A[J])
      swap(A[i], A[J])
  }
}
```

#Algo for Bubble sort:

/* where A is an Array with n element */

1. for i=0 to i<n-1-i Do the following
2. for J=i+1 to J<n-i Do the following
3. IF A[i] > A[J]
4. swap A[i], A[J]
5. End for loop 2
6. End For Loop 1
7. End.

Header (date)

Name	Experiment No.	scholar No.

LAB Assignment (1) (i) Explain the following Sorting Algorithm.

(A) Selection sort (B) Bubble sort.

(ii) Write down the Algorithm for both

(iii) Evaluate time & Space complexity for both

(iv) write down the code & attach screenshot of output.