

	Student ID
Experiment #1	Student Name
Date	States

Experiment Title: Analysis of Algorithm based on Arrays

Aim/Objective: To understand the concept and implementation of Basic programs on arrays.

**Description:** The students will understand and able to implement programs on Arrays.

**Pre-Requisites:** 

Knowledge: Arrays

Tools: Code Blocks/Eclipse IDE

Pre-Lab:

Calculate the time complexity of the following:

1. Algorithm linear\_search(arr, target):

for i in range(len(arr)):

if aπ[i] == target:

return i

return -1

· Procedure/Program: #Include Lstudioih> intlinear search (intaxx [], int size, inttager) for (inti=0; icsize; it t) {-- (nti)

if (art(i) == target)

freturni; -- (nti) tn. tn

inti

o(n)

Pata and Results: let add size is (n) 3 the

Rest case: - the element is the first element may

Time (omployity o())

woultrasm. The element is not present in the adju Time complexit orn as get-olar

· Analysis and Inferences: The livear search of an algorith meant in the array until it Finds the target

Best rase: O(1) : - Wosstraset - o(n)

Course Title Decises 1	
Course Code(s) 23C52305	
ACADEMIC YEAR: 202	4-25
Page 1	of 93

Experiment #: Date

> 2. Algor n = le

for i i fo

retur

Cou Cou



Experiment #1	Student ID
	Student Name
Date	•

2. Algorithm function(arr):

n = len(arr)

for i in range(n): -(n+1) for j in range(0, n-i-1):  $n(n^{2})$  U( 24 1) + J + SU + V+) o(nz)

if arr[j] > arr[j+1]: 'n (n) arr[j], arr[j+1] = arr[j+1],  $arr[j] \cap$ 

return arr;

- Procedure/Program: no of bubble -soxt(intass(], infos)

  Fox(int i=0; icn=1; i+1) & (nt)

  Fox(int i=0; icn=1; i+1) & (nt)

  if (ass(i) > ass(it)) & n-1

  ass(i) = ass(it); np ass[iti]=tempi syy
- . Data and Results: let input astay [134,25,12,27,11] Step-1: [25,34, 12/14,22] Step-2 [25 1121 34, 11,22] Ste P-3[82125111 34122] Step-4[1211120127,34] Step-5[1112,27,25,34]

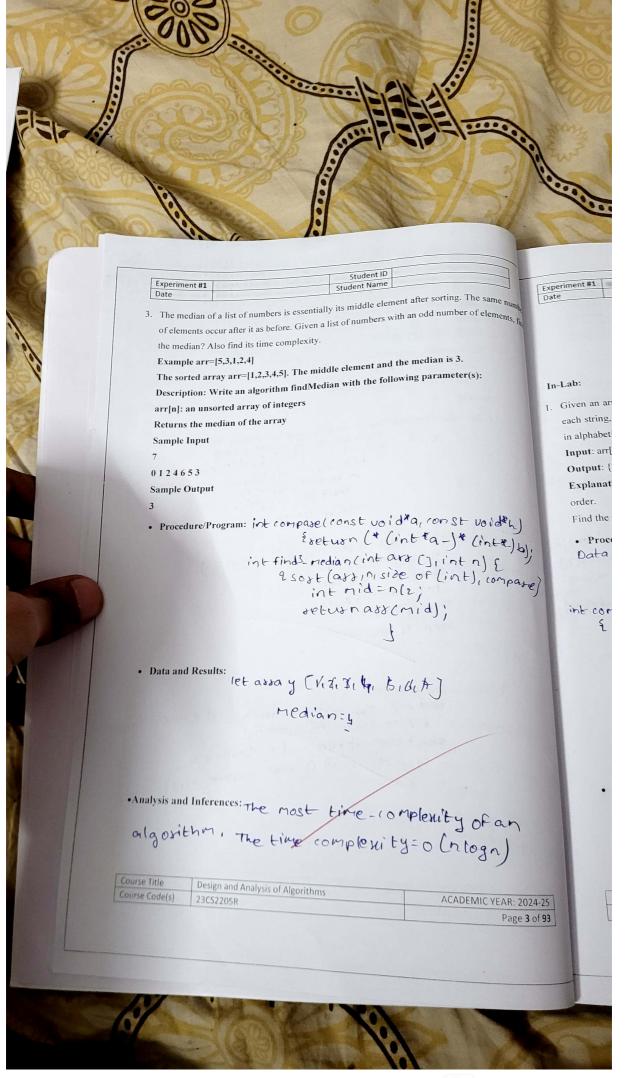
Analysis and Inferences:

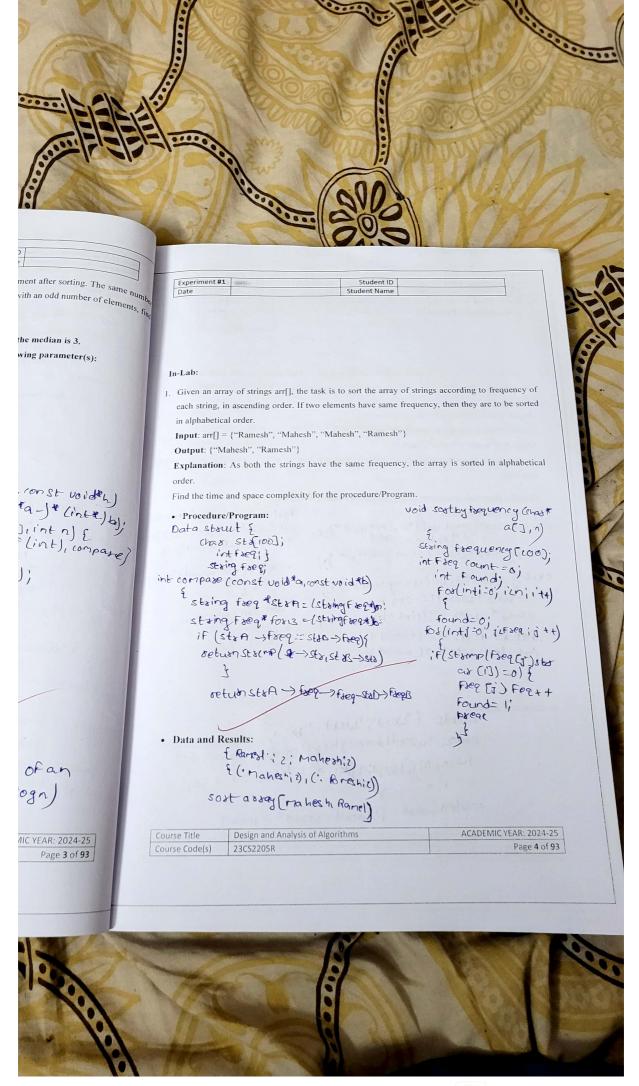
, Rest case. The array is dready sort make a Passes but no swap Postox out

worst case: it maken - it compassives

this result is Fine completely olne)

Course Title Design and Analysis of Algorithms Course Code(s) 23CS2205R	ACADEMIC YEAR: 2024-25 Page 2 of 93
Course	





Experiment #1	Student ID	
Date	Student Name	

## ·Analysis and Inferences:

2. Given an array of strings words [] and the sequential order of alphabets, our task is to sort the ar according to the order given. Assume that the dictionary and the words only contain lowere alphabets.

Input: words = {"word", "world", "row"},

Order= "worldabcefghijkmnpqstuvxyz"

Output: {"world", "word", "row"}

Explanation: According to the given order 'l' occurs before 'd' hence the words "world" will kept first.

Find the time and space complexity for the procedure/Program.

· Procedure/Program:

int compase (const void & , count void & b, if word [i] = word or int compase (const void & , count void & b, seturn order (cint) was

string order str B= (string order +) b; char words: StrA-) St chastorder = S(lint \*) order-map. int len listrien (word 1); int lenzistrien (word 2);

foslinti=o ii clentillenz string order \*str A= (string order \*1)a; -order (interestable) }

string order \*str B= (string order \*1) b;

} set-usn Lent-lenz;

· Data and Results:

mosq: { "wosq", wood , 11500) order: "wordldabe Fghij Klmnoparst vo wryz" fwio10:11,8:2,1:2,0:11,9:5,6:6,e: 7:e:a, 2008 fed word = ["world" word" 11/10000)

Course Title	Design and Analysis of Algorithms	ACADEMIC YEAR: 2024-25
Course Code(s)	23CS2205R	Page 5 of 93

Experiment #1	Student ID	
Date	Student Name	

· Analysis and Inferences:

## Post-Lab:

Evaluate the time complexity of pre-lab using master's theorem.

· Procedure/Program:

T(n) = at 
$$(n \mid n)$$
 + f(n) T = 3T  $(n \mid z)$  + n z  
where

 $q = 3$ 
 $n = size of an input f(n) = n$ 

as number of each supportable 1911 sub Problems f(n)= tost of the work done outside the recursive call which including the cost of dividing the Product

arl and by ale constaxts and f(n) is as asymp estically for

· Data and Results:

If 
$$f(n) = o(n \log b a - B)$$
 then
$$T(n) = o(n \log b a)$$

Course Title	Design and Analysis of Algorithms	ACADEMIC YEAR: 2024-25
Course Title		Page 6 of 93
Course Code(s)	23CS2205R	108000135

Experiment #1	Student ID	
Date	Student Name	

E F(n) = O(n logba) a the tens = o(n loga \* logn)

f(n) = aln log to are) the tens = o(Fm)

· Analysis and Inferences: a= number of each supposite subproses hote a= 11 and b=1 are constants and fin) is a sint freally

Sample VIVA-VOCE Questions (In-Lab):

- 1. What is the significance of analyzing both time complexity and space complexity whe evaluating algorithms?
- 2. How does the choice of data structures impact both time and space complexity in algorithm implementations? Give examples.
- 3. When comparing two algorithms with different time and space complexities, how do yo decide which one is more suitable for a particular problem or application?
- 4. What is the difference between best-case, average-case, and worst-case time complexity Provide examples.
- 5. In the context of sorting algorithms, compare the time and space complexities of algorithm like Quick Sort and Merge Sort. Which one is more time-efficient, and which one is more space-efficient?

Evaluator Remark (if Any): Marks Secured: 22 out of 50

Evaluator MUST ask Viva-voce prior to signing and posting marks for each experiment.

Course Title	Design and Analysis of Algorithms	ACADEMIC YEAR: 2024-25
Course Code(s)	23CS2205R	Page 7 of 93

