USN | | | | | | | | | | |

Internal Assessment Test 2 – May/June 2026

Scheme and Solution

Sub:	Analysis and Designs of Algorithms					Sub Code:	BCS401	Branch:	ISE
Date:	02-06-2026	Duration:	90 min's	Max Marks:	50	Sem/Sec:	IV / A, B, C		OBE

	Answer any FIVE FULL Questions	MARKS	CO	RBT
1.	Design Horspool's Algorithm for string Matching Apply Horspool algorithm too find pattern BARBER in the test JIM_SAW_ME IN_A_BARBERSHOP. Algorithm- 4 Explanation-6	10	CO3	L3

1. ## Horspool's Algorithm

It is the algorithm used for string matching of pattern text in Text string.

I/p : P[0... m-1] , T[0 ... n-1]

o/p : the first occurrence of the pattern

Algorithm

```
HORSPOOL (P, T)
1. Create the Shift Table
2. Initialize i = m-1
3. while i<n do
       K=0
       while K<m and P[m-1-K] = T[i-K] do
              K = K+1
       if K = m
              return i-m+1
       else:
              i+ = Shift (P[i])
4. return not found
```

P = BARBER

T = JIM_SAW_ME_IN_A_BARBERSHOP

Shift Table

Character	Shift
B	2
A	4
R	3
E	1
All others	6

T = JIM_SAW_ME_IN_A_BARBERSHOP

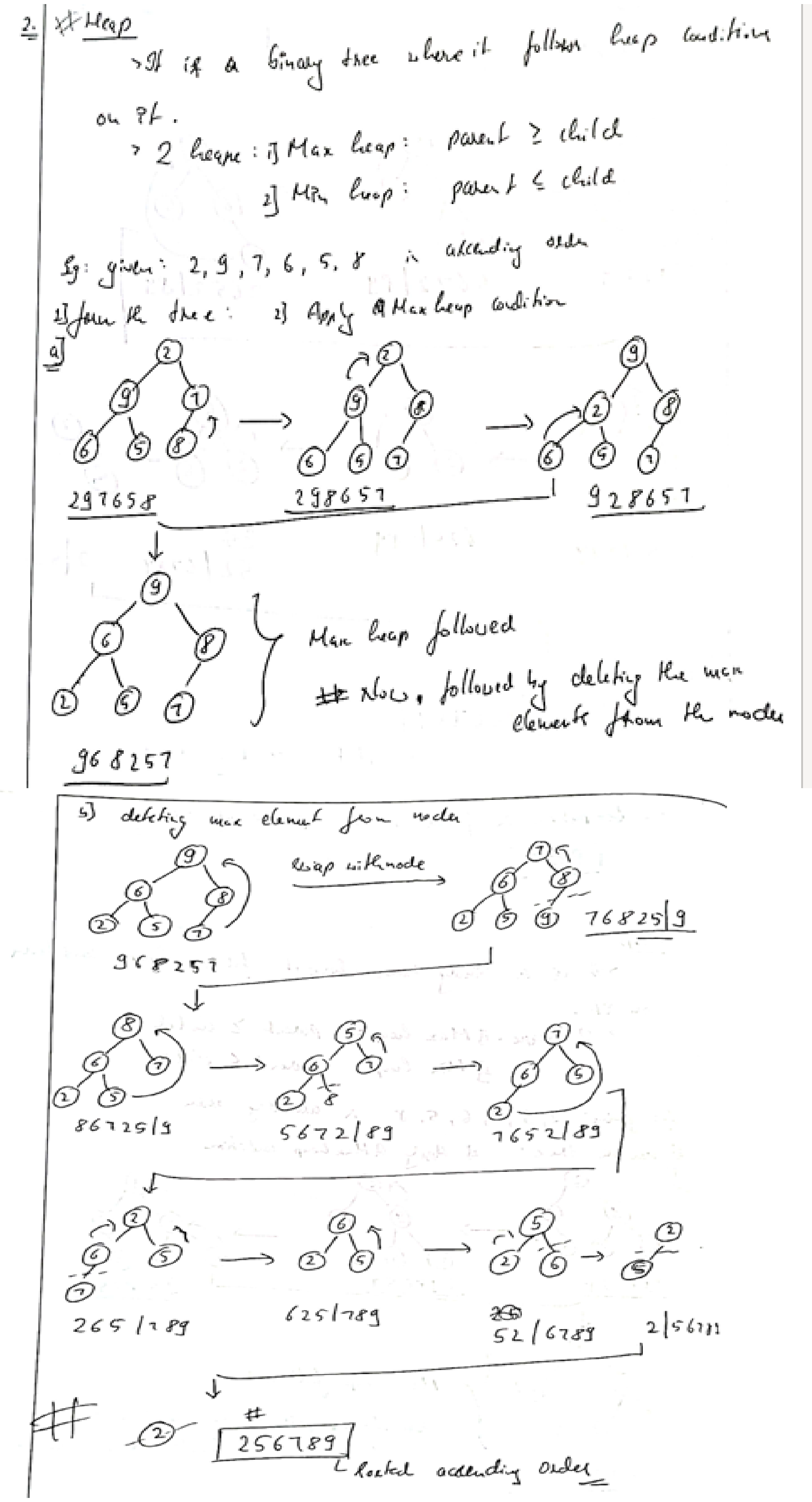
1. BARBER
2. BARBER
3. BARBER
4. BARBER
5. BARBER
6. BARBER ✓

Hence the string found in Text ✓

When ever the it is not matched it is incremented according to the shift Table

2. Define Heap. Apply heap sort to sort the list of numbers: 2, 9, 7, 6, 5, 8 in ascending order using array representation. — 10 — CO3 — L3

Definition-1
Algorithm-1
Steps & Explanation-8



3. What are Huffman trees? Construct the Huffman tree for the following data — 10 — CO4 — L3

Character	A	B	C	D	-
Probability	0.4	0.1	0.2	0.15	0.15

i) Encode the text ABAC-ABAD ii) Decode the code100010111001010

Definition- 1marks
Tree construction-4 marks
Coding Tab-3
Encode and decode- 2marks

3. Huffman tree

↳ It is a binary tree which follows the rule of huffman condition

↳ Parent is the sum of the nodes always

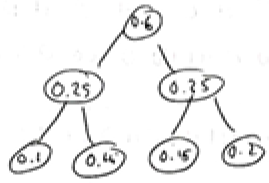
↳ helps to decode & encode the text/code.

Eg: given:

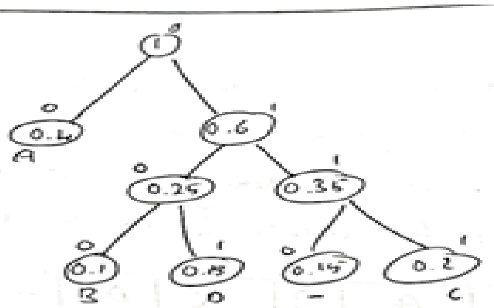
Char	A	B	C	D	_
Prob	0.4	0.1	0.2	0.15	0.15

1] Take smallest 2: B, D → Next smallest

2] As they both parents are less than the A, add both tree ~~and the~~



3] Add the tree to A:0.4



Hufman tree

According to the tree:

A = 0
B = 100
C = 111
D = 101
_ = 110

i) Encode: A B A C _ A B A D

0 100 0 111 110 0 100 0 101

= 01000111110 0100 0101

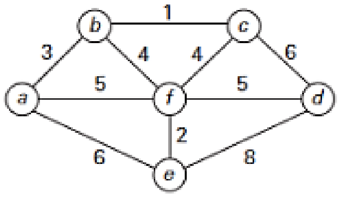
ii) Decode

1 0 0 | 0 | 1 0 1 | 1 1 0 | 0 | 1 0 1 | 0

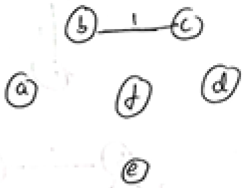
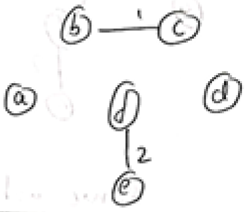
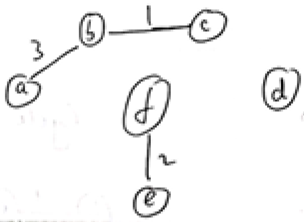
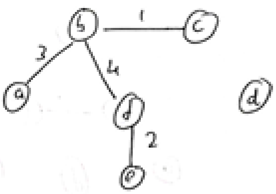
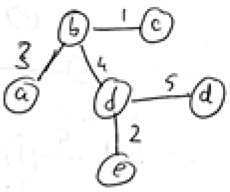
B A D _ A D A

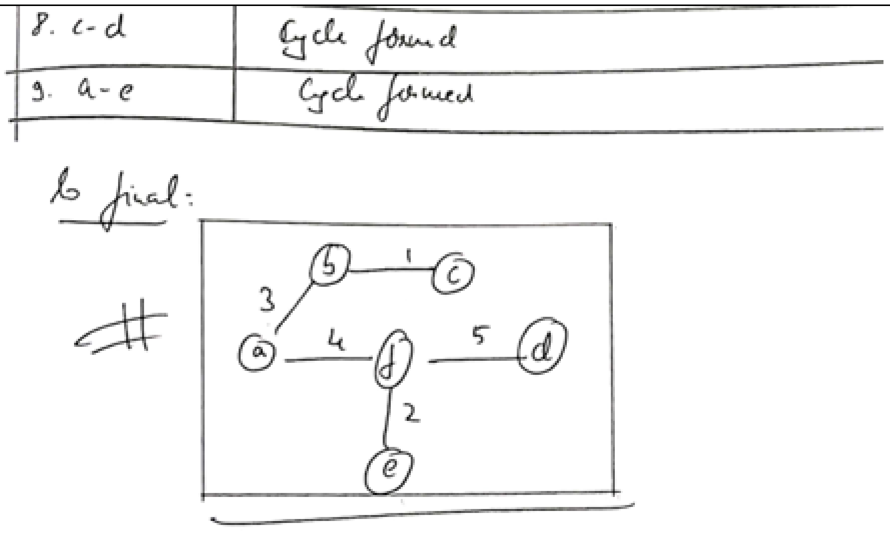
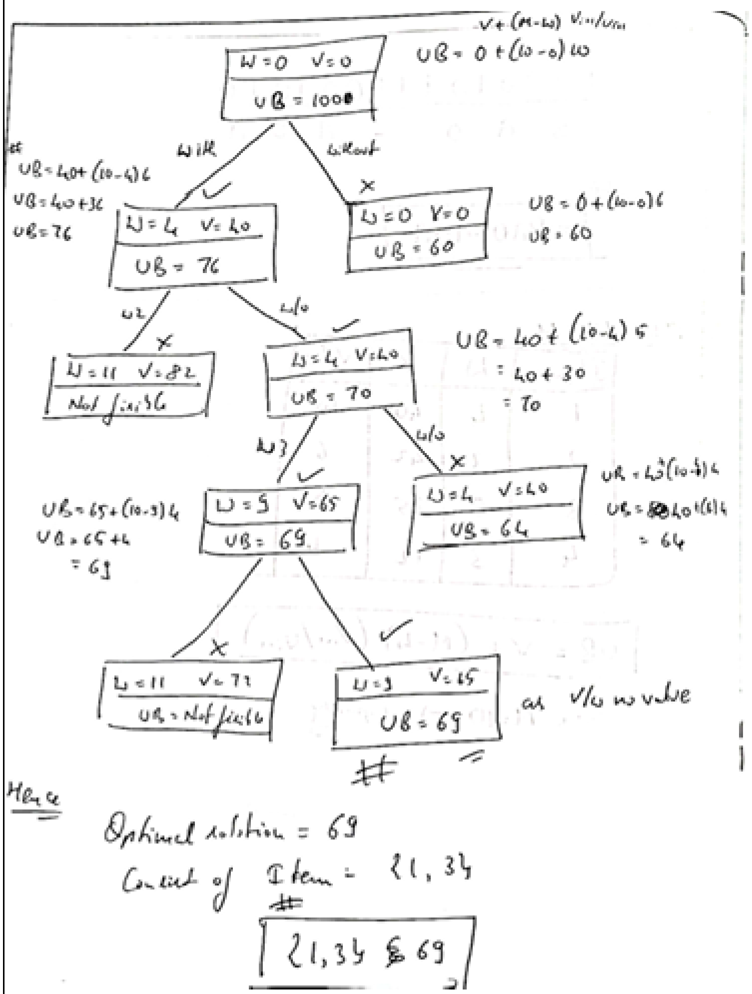
= BAD-ADA

4.	Construct minimum cost spanning tree using Kruskal's algorithm for the following graph. Algorithm- 2marks Solving- 7 marks MST- 1 marks	10	CO4	L3



4. Kruskals

Elements	Graph
1. b – c = 1	
2. f-e = 2	
3. a-b = 3	
4. b-f = 4	
5. c – f = 4	Same not change as ~~Greek~~ Cycle formation
6. a-f = 5	Cycle formation So, No
7. f – d = 5	

	 Minimum cost= 15			
5.	Solve the following instance of the knapsack problem by using branch and bound method. Capacity W = 10. Item: 1, 2, 3, 4 Weight: 4, 7, 5, 3 Value: \$40, \$42, \$25, \$12 Solving state space tree -10 	10	CO6	L3
6.	Explain the following terms: (i)P- Problems (ii) NP - Problems (iii) NP- Complete Problems (iv) NP- Hard Problems. Explaning (i)P- Problems – 2 marks (ii) NP – Problems – 2 marks (iii) NP- Complete Problems – 2 marks (iv) NP- Hard Problems. – 2 marks	10	CO5	L2

Item	1	2	3	4
Weight	4	7	5	3
Value	\$40	\$42	\$25	\$12

6. i) P- Problems

> It is deterministic solving problem with the polynomial time of $O(n^k)$
> It is the easy solvable problems as compared to other
> P- problem answer can be fetched easily.
> Used in: Merge sort,
Binary search etc
> Most easiest problems solvable in polynomial time

ii) NP- Problems

> NP → Not deterministic Polynomial time
> These are not deterministic in nature
> Are the hard problems of P.
> The solving takes time as it is hard
> But, the solved solution/answer are checked very quickly
> Used in: Travelling Sales Man problems
Graph colouring problems

iii) NP- Complete

> These are the NP-Complete problem only if
> a) It should be a NP problem
> b) The NP should be reducable to polynomial time
> Hard problem of NP
> Used in: SAT
3-SAT
> Hard NP problem to solve even to check

iv) NP- Hard Problems

> The problem should be reducable to polynomial time
> It may / maynot be a NP
> Need not have the verification
> Atleast has hard the NP- Complete
> Used in: Travels Sales Man (Optimization)
Job scheduling Optimization

CI CCI HOD