DAA-Exam#2-Prep-HW#2-Sol

Dynamic Programming:

1. Matrix-Chain Multiplication Problem: Given a chain of matrices $\langle A_1, A_2, \ldots, A_n \rangle$, where A_i has dimensions $\mathbf{p_{i-1}}$ x $\mathbf{p_i}$, fully parenthesize the product $A_1 \cdot A_2 \cdots A_n$ in a way that minimizes the number of scalar multiplications.

$m[i, j] = 0$, if $i = j$; $\min_{i \le k < j} \{m[i, k] + m[k+1, j] + p_{i-1}p_kp_j\}$ if $i < j$

a) Suppose we would like to compute the product of four matrices, A_1, A_2, A_3, A_4, whose dimensions are given below: What is the fastest way to compute the product $A_1 A_2 A_3 A_4$?

	A_1	A_2	A_3	A_4
Dimensions	30×1	1×40	40×10	10×25

Sol: [cost/k]

	A_1	A_2	A_3	A_4
A_1	0			
A_2		0		
A_3			0	
A_4				0

	A_1	A_2	A_3	A_4
A_1	0	1200/1	700/1	1400/1
A_2		0	400/2	650/3
A_3			0	10,000/3
A_4				0

1400, $A_1 ((A_2 A_3) A_4)$

b) Find an optimal parenthesization of a matrix chain product whose sequence of dimensions is (8, 10, 6, 11, 3, 35).

Sol:

A_1 (8 x 10), A_2 (10 x 6), A_3 (6 x 11), A_4 (11 x 3), A_5 (3 x 35)

	A_1	A_2	A_3	A_4	A_5
A_1	0	480/1	1008/2	618/1	1458/4
A_2		0	660/2	378/2	1428/4
A_3			0	198/3	828/4
A_4				0	1155/4
A_5					0

1458, $(A_1 (A_2 (A_3 A_4))) A_5)$

c) Find an optimal parenthesization of a matrix chain multiplication whose sequence of dimensions is (7, 10, 5, 16, 9, 22).

$\min_{i \le k < j} \{m[i, k] + m[k+1, j] + p_{i-1}p_kp_j\}$ **if** $i < j$

Sol:

A_1 (7 x 10), A_2 (10 x 5), A_3 (5 x 16), A_4 (16 x 9), A_5 (9 x 22)

	A_1	A_2	A_3	A_4	A_5
A_1	0	350/1	910/2	1385/2	2771/4
A_2		0	800/2	1170/2	2810/2
A_3			0	720/3	1710/4
A_4				0	3168/4
A_5					0

2771, (((A_1 A_2) (A_3 A_4)) A_5)

d) Find an optimal parenthesization of a matrix chain product whose sequence of dimensions is (6, 11, 7, 15, 3, 21).

Sol:

A_1 (6 x 11), A_2 (11 x 7), A_3 (7 x 15), A_4 (15 x 3), A_5 (3 x 21)

	A_1	A_2	A_3	A_4	A_5
A_1	0	462/1	1092/2	744/1	1122/4
A_2		0	1155/2	546/2	1239/4
A_3			0	315/3	756/4
A_4				0	945/4
A_5					0

1122, (A_1 (A_2 (A_3 A_4))) A_5)

2. Longest Common Subsequence (LCS): Given 2 sequences, $X = \langle x_1,\ldots,x_m\rangle$ and $Y = \langle y_1,\ldots,y_n\rangle$, find a common subsequence whose length is maximum. Subsequence need not be consecutive, but must be in order.

if (A[i] = B[j]) LCS[i,j] = 1 + LCS[i-1, j-1];

else LCS[i,j] = max(LCS(i-1, j), LCS(i, j-1));

a) Find LCS for strings X = ⟨longest⟩, Y = ⟨stone⟩. What are the LCS and its length?

one, 3

		l	**o**	**n**	**g**	**e**	**s**	**t**
	0	*0*	*0*	*0*	*0*	*0*	*0*	*0*
s	*0*	0	0	0	0	0	1	←1
t	*0*	0	0	0	0	0	1	2
o	*0*	0	1	1	1	1	1	2
n	*0*	0	1	**2**	←2	2	2	2
e	*0*	0	1	2	2	**3**	←3	←3

b) Find LCS for strings X = ⟨abcda⟩, Y = ⟨cdabac⟩. What are the LCS and its length?

abc,3; aba, 3; **cda, 3**

		a	**b**	**c**	**d**	**a**
	0	*0*	*0*	*0*	*0*	*0*
c	*0*	0	0	1	1	1
d	*0*	0	0	1	2	2
a	*0*	1	1	1	2	3
b	*0*	1	2	2	2	3
a	*0*	1	2	2	2	3
c	*0*	1	2	3	3	3

c) Find LCS for strings X = ⟨cdabac⟩, X = ⟨abcda⟩. What are the LCS and its length?

		c	**d**	**a**	**b**	**a**	**c**
	0	*0*	*0*	*0*	*0*	*0*	*0*

a	*0*	0	0	1	1	1	1
b	*0*	0	0	1	2	2	2
c	*0*	1	1	1	2	2	3
d	*0*	1	2	2	2	2	3
a	*0*	1	2	3	3	3	3

abc, 3

3. Optimal Binary Search Trees: Given sequence $K = k_1 < k_2 < \cdots < k_n$ of n sorted keys, with a search probability p_i for each key k_i. Want to build a binary search tree (BST) with minimum expected search cost.

a) Build all unique BSTs with keys [4, 5, 6].

Check lecture ppt

b) What is the number of possible unique BSTs for 7 keys? Use the given 6 keys of Catalan sequence.

The number of keys:

n = 1, # BST 1

n = 2, #BST 2

n = 3, #BST 5 (2 + 1 + 2)

n = 4, #BST 14 (5 + 4 + 5)

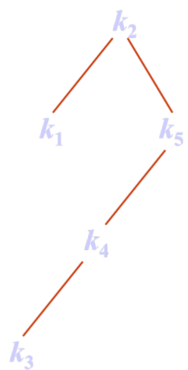
n = 5, #BST 42 (14 + 5 + 4 + 5 + 14)

n = 6, #BST 132 (42 + 14 + 10 + 10 + 14 + 42)

132 + 42 + 28 + 25 + 28 + 42 + 132 = 429

14C7/8 = 429

c) Consider 5 keys with these search frequencies: $f_1 = 3$, $f_2 = 4$, $f_3 = 7$, $f_4 = 2$, $f_5 = 10$
What is the expected total search cost for the following BST (depth starts from 1 (k_2 is 1, k_3 is 4))?



1 * 4 + 2 * (3 + 10) + 3 * 2 + 4 * 7 = 64

d) Consider the following 4 keys with the search frequencies. What is Optimal Binary Search Tree? Show your work sequence using the given table and show the optimal BST.

Key	1	2	3	4
Key Value	10	20	30	40
Frequency	4	2	6	3

1 =< i =< j+1 =< n

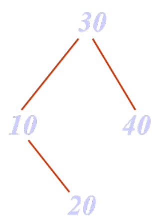
$C[i,i] = P_i$

$C[i,i-1] = 0$

$C[i,j] = \min \{C[i,k-1] + C[k+1,j] + w(i,j)\}$ for k =i to j

w(i,j) = sum of F from i to j

i/j	0	1	2	3	4
1	0	4	8/1	20/3	26/3
2		0	2	10/3	16/3
3			0	6	12/3
4				0	3
5					0



4. Fibonacci Numbers:

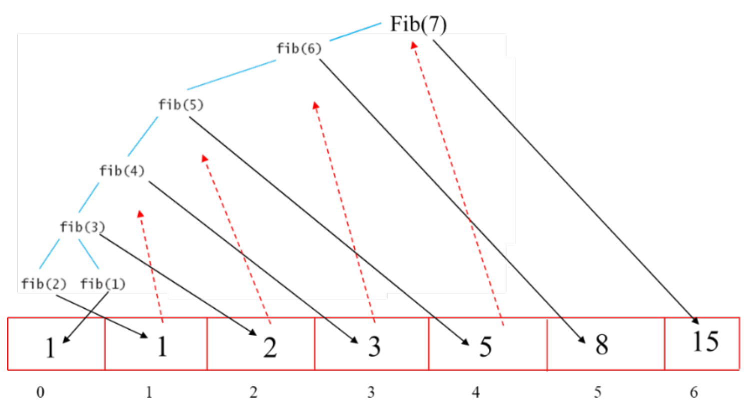
a) Computing the nth Fibonacci number recursively using the following algorithm, what is the time complexity?

```
int Fib(int n)
{
    if (n <= 1)
        return 1;
    else
        return Fib(n - 1) + Fib(n - 2);
}
```

$2T(n\text{-}1) + 1 >= T(n) = T(n\text{-}1) + T(n\text{-}2) + 1 >= 2T(n\text{-}2) + 1$

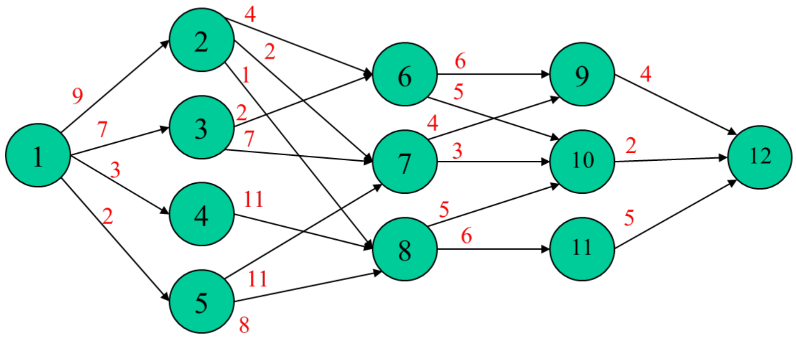
$O(2^n)$ and $\Omega(2^{n/2})$ ← linear master method

b) What is fib(7)? Show the computing process by using memorize method.



5. The shortest path in multistage graphs:

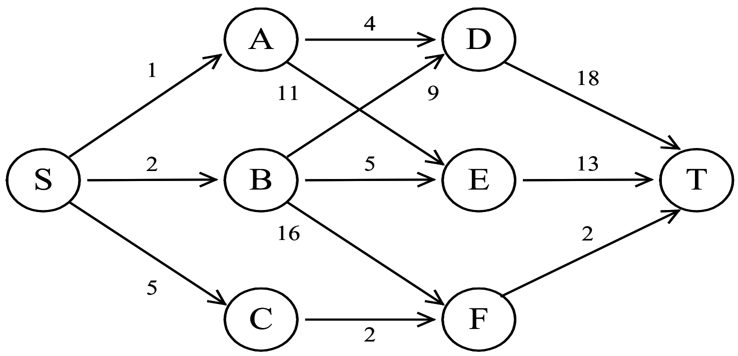
a) For the given multistage graph, find the shortest path by using dynamic programming.



Cost(i, j) = min {C(j,l) + Cost(i+1,l)} i:stage, j: vertex #

1	2	3	4	5	6	7	8	9	10	11	12
16	7	9	18	15	7	5	7	4	2	5	0
2/3	7	6	8	8	10	10	10	12	12	12	12

b) For the given multistage graph, find the shortest path by using dynamic programming.



S	A	B	C	D	E	F	T
9	22	18	4	18	13	2	0
C	D	E/F	F	T	T	T	T

6. 0-1 Knapsack problem: Given a knapsack with maximum capacity W, and a set S consisting of n items. Each item i has some weight w_i and benefit value b_i (all w_i and W are integer values).

Problem: How to pack the knapsack to achieve maximum total value of packed items (each item must be entirely accepted or rejected)?

$$V[k,w] = \begin{cases} V[k-1,w] & \text{if } w_k > w \\ \max\{V[k-1,w], V[k-1,w-w_k]+b_k\} & \text{else} \end{cases}$$

Let i=n and k=W

if $V[i,k] \neq V[i-1,k]$ then mark the i^{th} item as in the knapsack; $i = i-1$, $k = k-w_i$

else $i = i-1$;

a) For the given n = 4 (# of elements), W = 5 (max weight), Elements (weight, benefit): (2,3), (3,4), (4,5), (5,6), find the maximum possible value that can be carried in the knapsack and identify the items.

i\W	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	3	3	3	3
2	0	0	3	4	4	7
3	0	0	3	4	5	7
4	0	0	3	4	5	7

b) For the given n = 4 (# of elements), W = 5 (max weight), Elements (weight, benefit): (2,3), (3,4), (1,5), (5,6), find the maximum possible value that can be carried in the knapsack and identify the items.

i\W	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	3	3	3	3
2	0	0	3	4	4	7
3	0	5	5	8	9	9
4	0	5	5	8	9	9

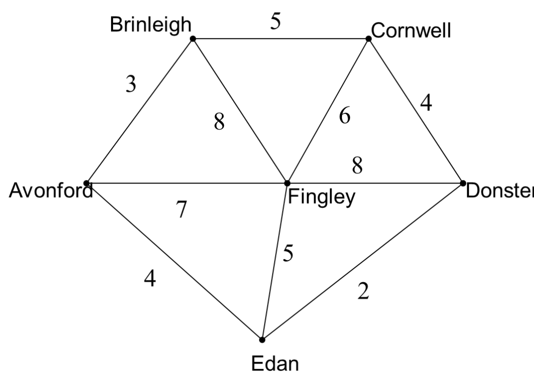
b) For the given n = 4 (# of elements), W = 10 (max weight), Elements (weight, benefit): (5,10), (4,40), (6,30), (3,50), find the maximum possible value that can be carried in the knapsack and identify the items.

i\W	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	10	10	10	10	10	10
2	0	0	0	0	40	40	40	40	40	50	50
3	0	0	0	0	40	40	40	40	40	50	50
4	0	0	0	50	50	50	50	90	90	90	90

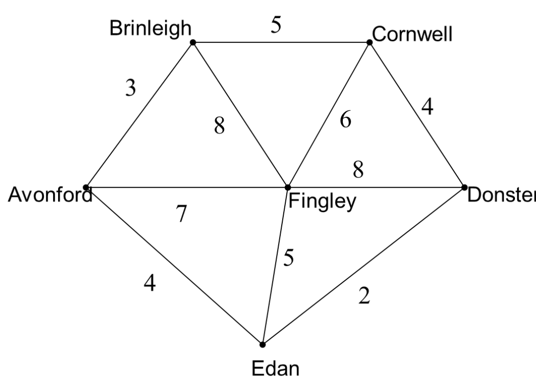
Greedy Algorithms:

1. Minimum Spanning Trees:

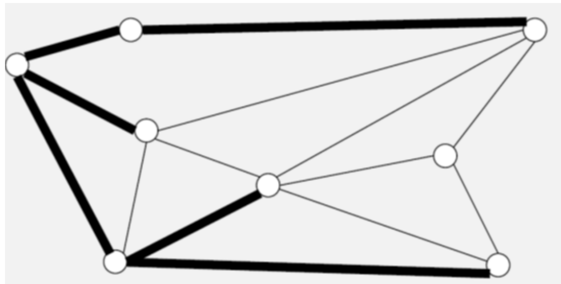
a) A cable company want to connect five villages to their network which currently extends to the market town of Avonford. What is the minimum length of cable needed? Show your work by using Kruskal's Algorithm.



b) A cable company want to connect five villages to their network which currently extends to the market town of Avonford. What is the minimum length of cable needed? Show your work by using Prim's Algorithm.



c) Why the following is not a spanning tree? Not Spanning



2. Huffman encoding:

a) To send the following characters, how many bits are necessary by using Huffman encoding? Show the Huffman encoding design.

aaaaaaaaaaaabbbeeeww (n = 20, using ASCII code (8 bits))

	freq	code
a	12	0
b	3	10
e	3	110
w	2	111

Data: 12 + 6 + 9 + 6 = 33 bits

Table: 8 * 4 + 9 = 41 bits

Total = 74 bits

3. Job sequencing with deadline:

a) Schedule the following job to maximize the jobs meeting the deadline.

	J1	J2	J3	J4	J5	J6	J7
Jobs	35	30	25	20	15	12	5
deadlines	3	4	4	2	3	1	2

1	2	3	4
J4	J3	J1	J2

4. The Fractional Knapsack Algorithm: One wants to pack n items in a luggage: the ith item is worth vi dollars and weighs wi pounds. Maximize the value but cannot exceed W pounds

vi , wi, W are integers. Fractions of items can be taken.

a) For the given items with a jar with 10 ml jar, solve a fractional knapsack problem.

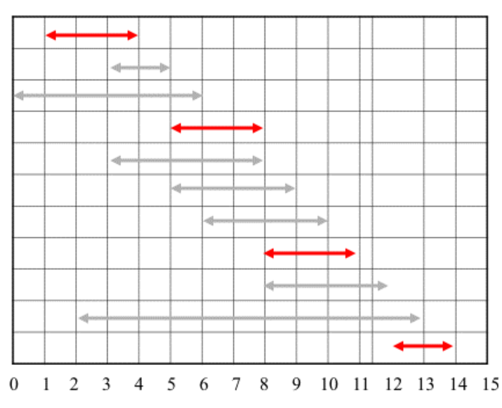
	1	2	3	4	5
Weight	4 ml	8 ml	2 ml	6 ml	1 ml
Benefit	$12	$32	$40	$30	$50

Value per ml 3 4 20 5 50

50 (5: 50 *1) + 40 (3: 20 *2) + 30 (4: 5 * 6) + 4 (2: 4 * 1) = $124

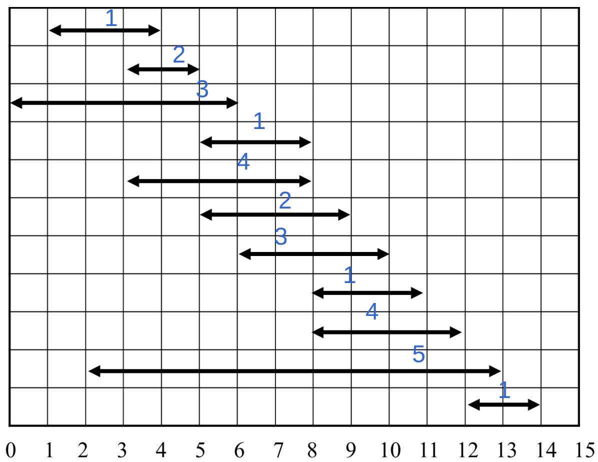
5. Conference Scheduling Problem: A given set of activities (with start and finish times), What is the maximum number of mutually compatible activities that can be completed? Show your greedy templates and algorithms.

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14



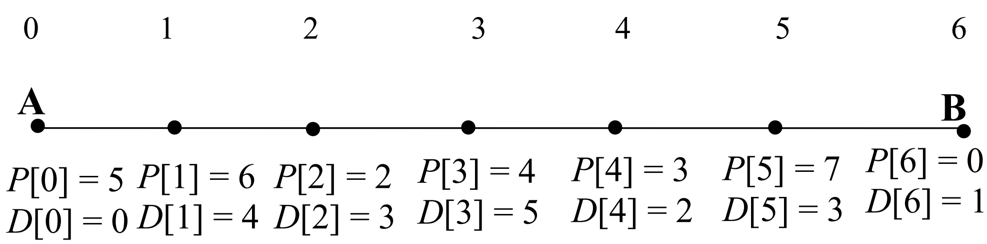
6. Interval Partitioning Problem: find minimum number of classrooms to schedule all lectures so that no two occur at the same time in the same room.

i	1	2	3	4	5	6	7	8	9	10	11
s_i	1	3	0	5	3	5	6	8	8	2	12
f_i	4	5	6	7	8	9	10	11	12	13	14



The depth of a set of open intervals is the maximum number that contain any given time. 5 for this question.

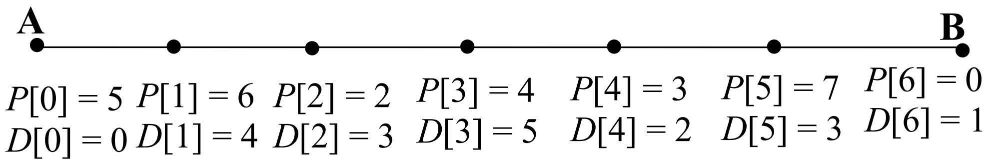
7. Selecting Breakpoints: Make as few refueling stops as possible on the road trip along fixed route (given fuel capacity = L= 8 and gas station map).



Stop at 2 and 4; Greedy algorithm. Go as far as you can before refueling.

8. The gas station scheduling problem: the amount of gas at each station to minimize the total cost from A to B.

Suppose we plan to drive in a car from city A to city B. Along the high way, we will go through n gas stations, labeled with 1, 2, …, n. For convenience, let station 0 be located in city A and station n+1 in city B. The distance from station i-1 to station i is given in an array D[i] which is known in advance. Let the gas price at station i be P(i), $0 \le i \le n+1$ which is also known. For simplicity, we assume P(i) has been converted to the dollar amount per each mile for the car we use. You may assume P(n+1) = 0. An example is given below.



Let L (miles) be the distance a car can run with full tank of gas. It is assumed that in any L miles, there is at least one station and at most k stations, where k is a constant.
Now, we want to compute the amount of gas that needs be added (in terms of miles) at each station such that the total cost is minimized to drive the car from city A to city B, assuming the gas tank is empty initially.
L is 8, find the minimum total cost from A to B.

- The total cost is 7×5 + 8×2 + 3×3 = 60
- Greedy algorithm.

Within L miles, station u is the first such that P[u] <= P[i]. Add gas enough to arrive to station *u*.

Within L miles, station u has the lowest price but P[u] > P[i]. Add as much gas as possible to make the tank full. $G(i) = L - R(i)$ gas.

9. Schedulability Problem: A set of tasks is said to be schedulable under a given set of constraints, if a schedule exists for that set of tasks and constraints.

Scheduling for two periodic tasks:

	First Arrival	Deadline	Period	Execution Time
T1	0	5	5	1
T2	3	7	7	1

Hints: $2^{1/2} = 1.41$, $2^{1/3} = 1.25$, $2^{1/4} = 1.19$

Necessary and sufficient conditions for RM scheduling on a single processor and n tasks:

$$\mu = \sum_{i=1}^{n} \frac{c_i}{p_i} \leq 1 \qquad\qquad \mu = \sum_{i=1}^{n} \frac{c_i}{p_i} \leq n(2^{1/n} - 1)$$

1. [2]Show the Utilization Bound test for the above tasks. State if the RM schedulability is conclusive or inconclusive.

n = 2, n($2^{1/2}$ -1) = 0.82; 1/5 + 1/7 = 0.342 < 0.82 → Conclusive

2. [2] Show the schedule using Rate-Monotonic (RM) scheduling algorithm up to 25 time steps using the given table.

T1	1					1					1					1					1				
T2				2								2						2							

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

3. [1] For the above scheduling scenarios, if preemptions happen, state how many preemptions hap pened and indicate the time of each preemption.

NO preemption

4. [1] For the above scheduling scenarios, is there any process missing the deadline? If yes, please indicate the process and the time when it happened.

NO deadline miss

5. [2] Show the schedule using Earliest Deadline First (EDF) algorithm up to 25 time steps (if processes are tied in deadline, no preemption will be invoked and the shorter period process has the higher priority).

T1	1					1					1					1					1				
T2				2								2						2							

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

6. [1] For the above scheduling scenarios, if preemptions happen, state how many preemptions happened and indicate the time of each preemption.

NO preemption

7. [1] For the above scheduling scenarios, is there any process missing the deadline? If yes, please indicate the process and the time when it happened.

NO deadline miss

Scheduling for two periodic tasks:

	First Arrival	Deadline	Period	Execution Time
T1	0	5	5	2
T2	3	7	7	4

Hints: $2^{1/2} = 1.41$, $2^{1/3} = 1.25$, $2^{1/4} = 1.19$

Necessary and sufficient conditions for RM scheduling on a single processor and n tasks:

$$\mu = \sum_{i=1}^{n} \frac{c_i}{p_i} \le 1 \qquad\qquad \mu = \sum_{i=1}^{n} \frac{c_i}{p_i} \le n(2^{1/n} - 1)$$

8. [2] Show the Utilization Bound test for the above tasks. State if the RM schedulability is conclusive or inconclusive.

n = 2, $n(2^{1/2} - 1) = 0.82$; 2/5 + 4/7 = 0.97 > 0.82 → inconclusive, but 0.97 < 1 → schedulable

9. [2] Show the schedule using Rate-Monotonic (RM) scheduling algorithm up to 25 time steps using the given table.

T1	1	1				1	1				1	1				1	1				1	1			
T2				2	2			2	2				2	2	2			2	2	2			2		

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

10. [1] For the above scheduling scenarios, if preemptions happen, state how many preemptions hap pened and indicate the time of each preemption.

Preemption on 4 ~ 5, 19 ~ 20; two preemptions

11. [1] For the above scheduling scenarios, is there any process missing the deadline? If yes, please indicate the process and the time when it happened.

T2 miss deadline on its second period

12. [2] Show the schedule using **Earliest Deadline First (EDF)** algorithm up to 25 time steps (if proce sses are tied in deadline, no preemption will be invoked; the shorter period process has the highe r priority).

T1	1	1						1	1		1	1					1	1					1	1	
T2				2	2	2	2						2	2	2	2			2	2	2	2			

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5

13. [1] For the above scheduling scenarios, if preemptions happen, state how many preemptions hap pened and indicate the time of each preemption.

NO preemption

14. [1] For the above scheduling scenarios, is there any process missing the deadline? If yes, please indicate the process and the time when it happened.

NO deadline miss