Roll Number:_______________________

Thapar Institute of Engineering & Technology

Department of Computer Science and Engineering

END SEMESTER EXAMINATION

B. E. **(2nd Yr. COE/CSE)**
27th May, 2024
Monday, Time-9:00 To 12:00 PM
Time: 3 Hours
Max Marks: 60 (Weightage: 40)

Course Code: UCS415
Course Name: Design and Analysis of Algorithms
Name of Faculty: Tarunpreet Bhatia, Sumit Sharma, Manisha Singh, Swati Sharma, Mansi Sharma, Shubhra Dwivedi, Shruti Aggarwal

Note: Attempt any 4 questions. Answer all sub-parts of each question at one place. Do mention Page No. of your attempt at front page of your answer sheet. Assume missing data (if any).

Q.1

(a) You are a team member responsible for enhancing the efficiency and reliability of an urban bike-sharing system. The city is represented as a graph, with bike stations serving as vertices and bike lanes connecting them as edges. Your primary objective is to design an algorithm for bike lane maintenance that ensures every bike lane is traversed exactly once by a maintenance crew, which conducts station checks for maintenance needs and redistributes bikes if necessary. (5)

(b) Suppose you are flying by Vistara Airlines from New Delhi to Tokyo, and you know the weight restriction for cabin baggage, which is 7 kg. Additionally, the items you carry are unique, each with different weights and values, as shown in **Table 1**. However, the combined weight of all the items exceeds 7 kg, so you must inform the customs officer to retain the items with less weight but higher value. Each item is either allowed to be taken in baggage or not carried. Your primary objective is to carry the items with maximum value in your cabin baggage without surpassing the baggage limit. (7+3)

Table 1

Items	Smartphone	Tablet	Power bank	Laptop
Weights	3	4	2	5
Values	40	50	10	70

i) Which items will be carried in your cabin bag? Write the recursive equation and show your calculations with the help of dynamic programming.

ii) How would your answer vary if greedy programming is used to select the items for filling your cabin bag? Justify your answer with proper reasoning.

Solution

(a). Fluery's Algorithm or Hierholzer's algorithm **(5 Marks only if properly written)**
Note: If the Euler circuit or Euler path discussed then **only 1 mark**

(b). (i) for dynamic 0/1 knapsack table: **3 marks** (for incorrect entries marks deducted)
recursive equation: **2 marks (if only 1 part is written: 1 mark)**
1 mark for showing steps of calculation
1 mark for items carried
(ii) **1 mark** for Profit/wt value
2 mark for the answer with the reason

wt	3	4	2	5
values	40	50	10	70

⇓

	I_3	I_1	I_2	I_4
wt	2	3	4	5
value	10	40	50	70

⇓

value	wt	i \ w	0	1	2	3	4	5	6	7
		0	0	0	0	0	0	0	0	0
10	2	1	0	0	10	10	10	10	10	10
40	3	2	0	0	10	40	40	50	50	50
50	4	3	0	0	10	40	50	50	60	90
70	5	4	0	0	10	40	50	70	70	(90)

(90) ⇒ optimal sol?

wt: 3, 4 → (I_1, I_2)

0/1 Knapsack using DP

$$V[i,w] = \begin{cases} V[i-1,w] & \text{if } w_i > w \\ \max\{V[i-1,w],\ V[i-1,w-w_i]+b_i\} & \text{else} \end{cases}$$

(Smartphone & Tablet)

0/1 Knapsack using Greedy

wt	5	3	4	2
value	70	40	50	10
v/wt	14	13.3	12.5	5

Select item with wt. 5
value = 70
Remaining capacity = 2
∵ fraction not allowed
∴ select item with wt. 2
value = 70 + 10 = (80) — not optimal
Item selected (I_4, I_3) (Laptop: Power Bank)

Another table:

0	0	0	0	0	0	0	0	0
1	0	0	0	40	40	40	40	40
2	0	0	0	40	50	50	50	90
3	0	0	10	40	50	50	60	90
4	0	0	10	40	50	50	70	90

Q.2 (a) You are organizing a fundraising event for your local charity. You have a list of potential donors, each willing to contribute a certain amount of money. The potential donor contributions are {5K, 6K, 10K, 11K, 16K}. However, you want to find out all the possible combinations of donors whose contributions sum up to a target amount of 21K to efficiently reach your fundraising goal. Show the state space tree to find all possible combinations of donors whose contributions sum up to the target amount using the backtracking technique by clearly specifying the pruning conditions on the state space tree. (8)

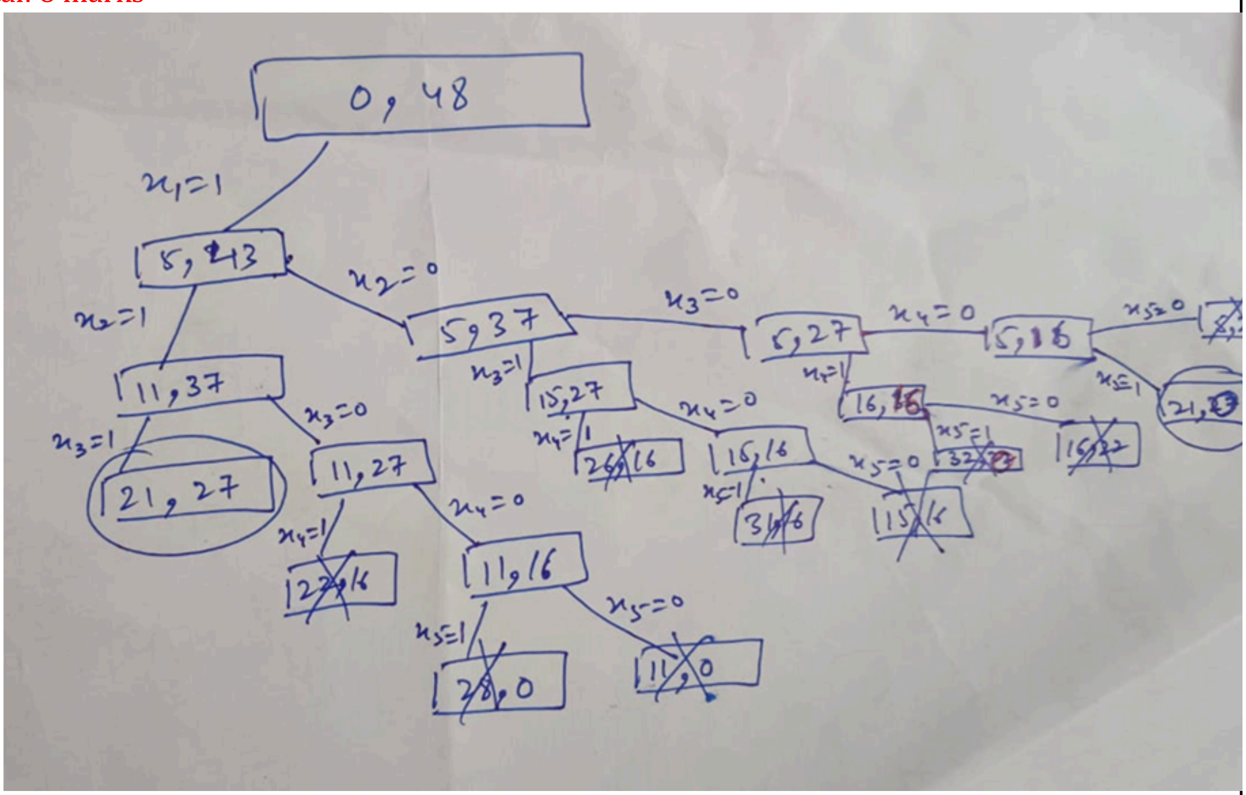
(b) Given a wall of length W and two shelves of length m and n, we are tasked with fitting the wall of length W with shelves of length m and n so that the space left empty (which can't be filled with shelf) is to be minimized, and if possible the solution having larger number of longer shelves is preferred as longer shelves are the cheaper ones. However, cost is still secondary in our adventure of minimizing the cost, we should be more worried about minimizing the empty space (if possible it should be zero). Design a greedy algorithm for this problem. (3)

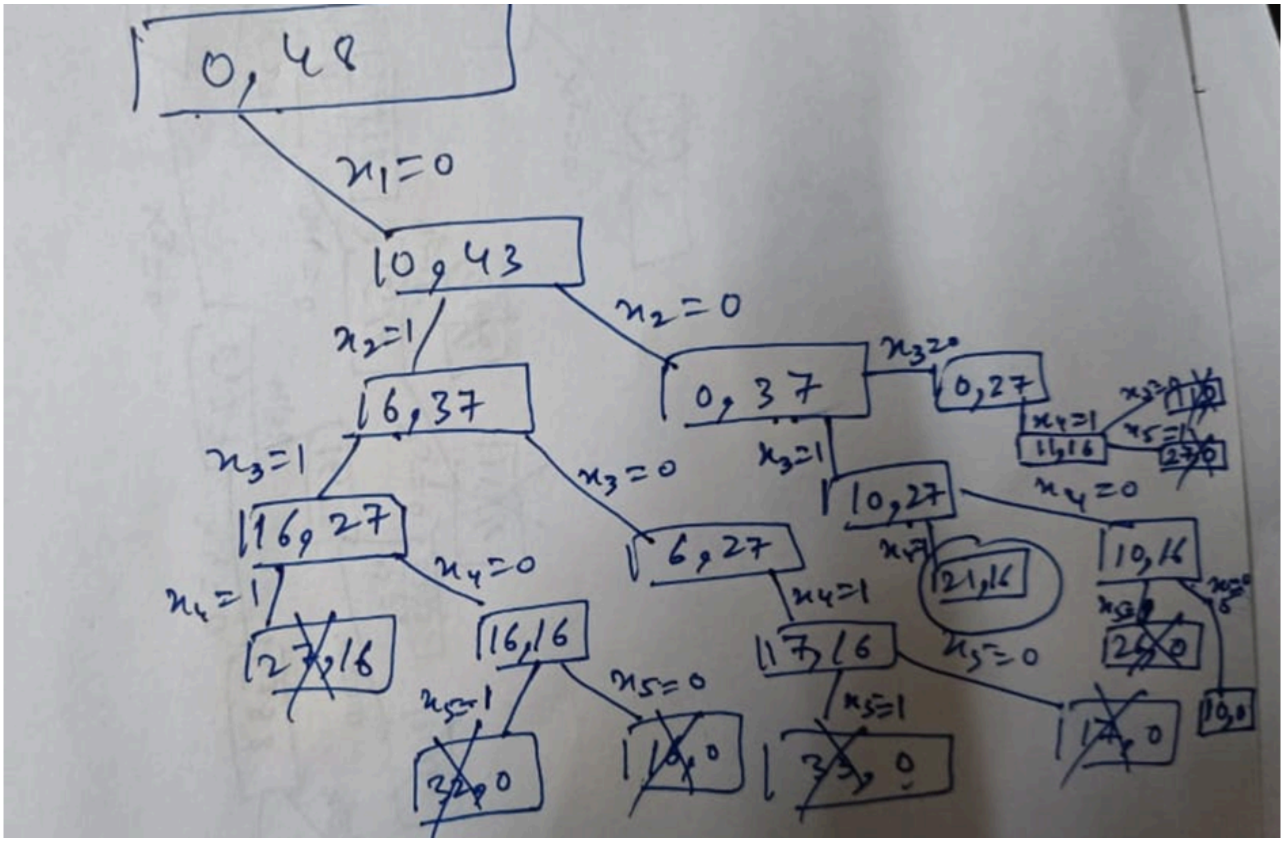
(c) Differentiate between Las Vegas and Monte Carlo randomized algorithms with the help of an example. (4)

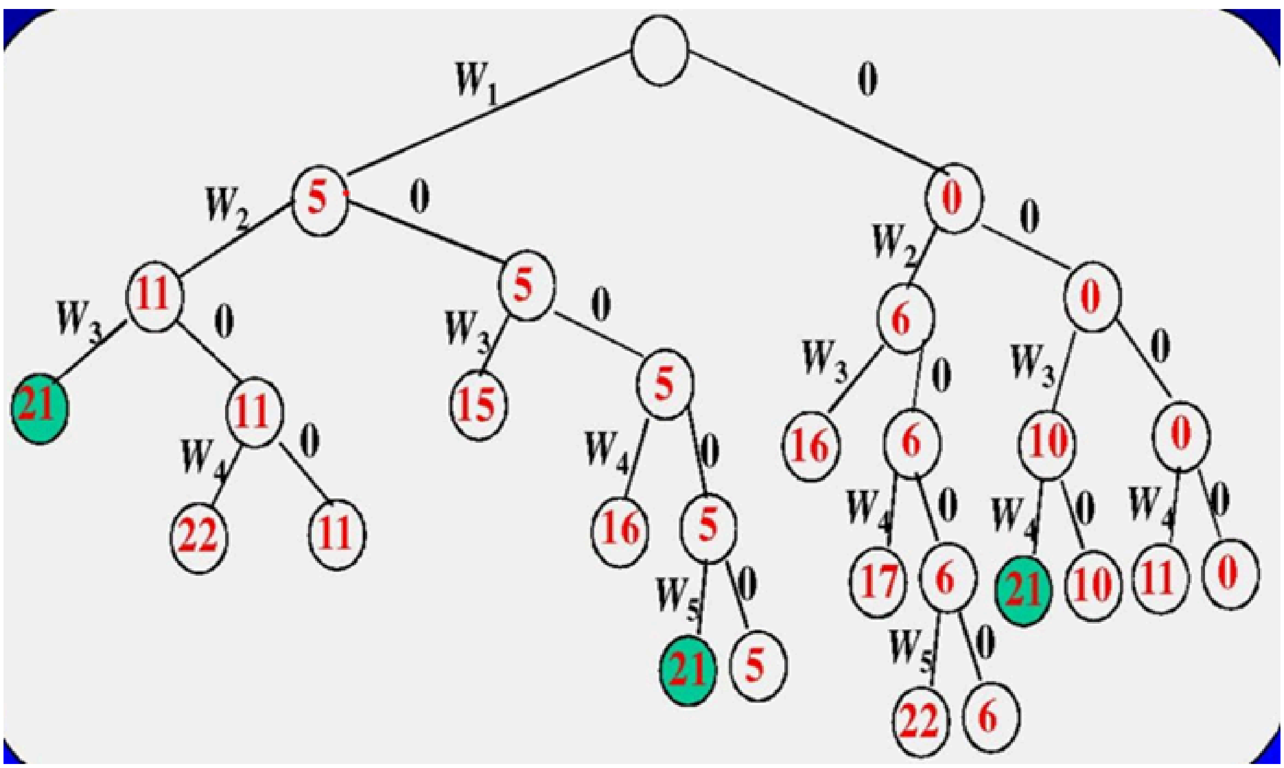
Solution:

a) 6 marks for state space tree, 1 mark showing pruning conditions, 1 mark to print the answer (there will be 3 patterns, so 1st answer- 0.5 marks and 2nd answer -0.5 marks, all three answers written- 1 mark.)

Total: 8 marks







b) Step 1: sort based on the length – 1 mark, Step 2: applied iteration – 1 mark, Step 3: Repeat till no space left – 1 mark

Total: 3 marks

1) Sort the lengths of the shelves in descending order. (1)

2) Initialize two variables to keep track of remaining space on the wall & no. of shelves placed.

3) Iterate through the sorted shelves length.

   i) Place shelves of longest length without exceeding the space (2)

   ii) Update remaining space & no. of shelves.

4) Repeat ~~step~~ step 3 for next longest shelf length until there is no space left. (3)

c) 3 marks for main differences and 0.5 marks each for example

Las Vegas	Monte Carlo
① Produce a correct result but their running time can vary depending on the randomness.	① It have fixed running time, but they produce a correct result with certain probability.
② eg. Randomized Quicksort	② eg Karger Min Cut.

Q.3 (a) Imagine a warehouse floor laid out in a grid pattern, awaiting the deployment of N autonomous robots assigned to collect items from various locations and deliver them to designated destinations. The challenge here is to ensure that no two robots collide or interfere with each other's paths while navigating the warehouse. The two robots collide if they occupy the same row, column, or diagonal. Each cell in the grid represents a possible position for a robot. To tackle this logistical puzzle, you decide to use the Backtracking approach to find a valid arrangement for the N robots within the warehouse. Design a pseudo code or algorithm for this problem using the Backtracking approach. Also, apply your algorithm to find one of the possible solutions for N = 4. You need to show the function calls while applying your algorithm. What will be the worst-case time complexity of your algorithm? (10)

Solution: 5 marks (pseudo code) + 4 marks (function call)

Place (k, i) returns a Boolean value that is true if the kth queen can be placed in column i. It tests both whether i is distinct from all previous costs x_1, x_2,....x_{k-1} and whether there is no other queen on the same diagonal.

Using place, we give a precise solution to then n- queens problem.

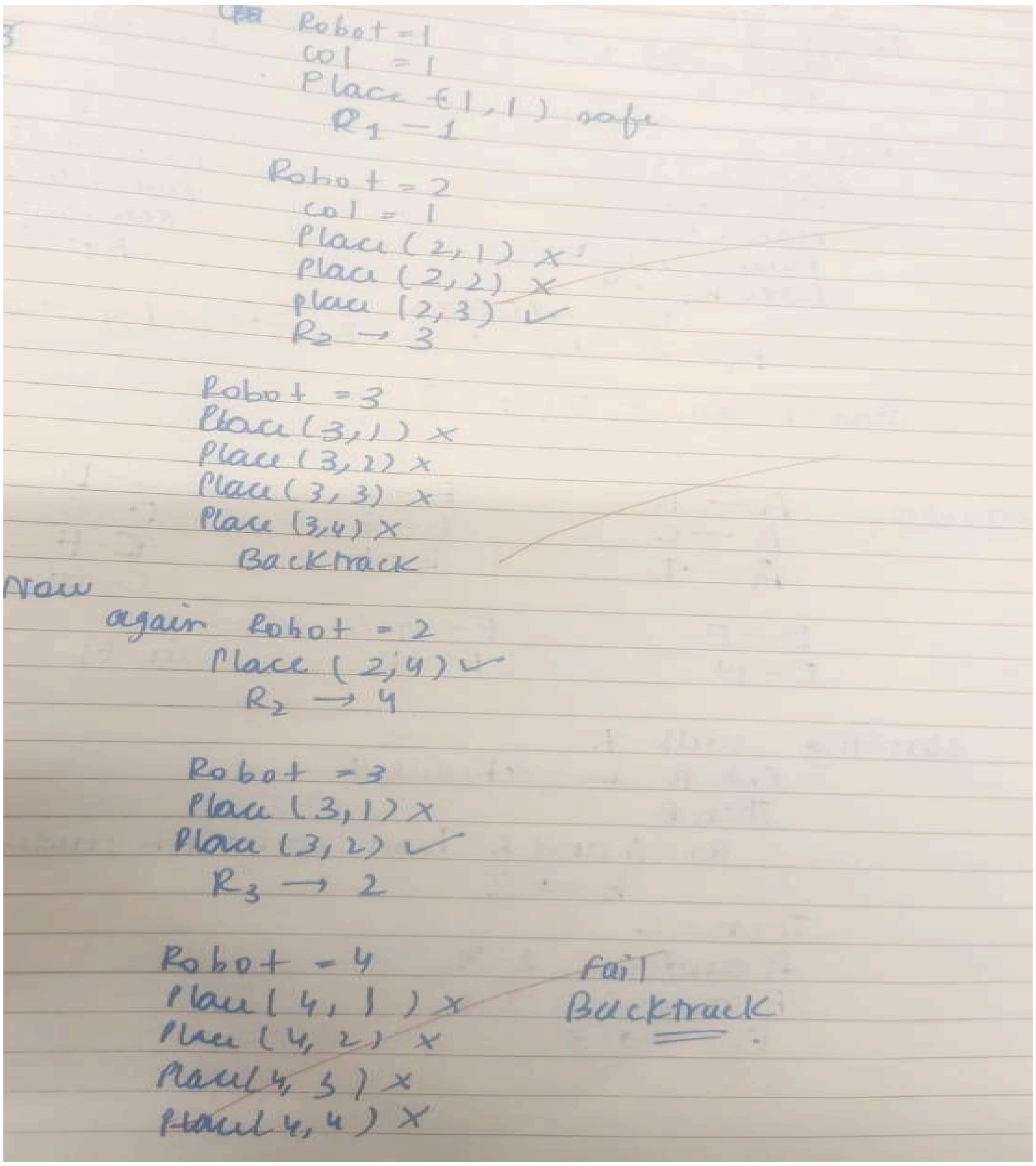
```
Place (k, i)
 {
  For j ← 1 to k - 1
   do if (x [j] = i)
    or (Abs x [j]) - i) = (Abs (j - k))
  then return false;
  return true;
}
```

Place (k, i) return true if a queen can be placed in the kth row and ith column otherwise return is false.

x [] is a global array whose final k - 1 values have been set. Abs (r) returns the absolute value of r.

```
N - Queens (k, n)
{
 For i ← 1 to n
    do if Place (k, i) then
 {
```

```
x [k] ← i;
if (k ==n) then
  write (x [1....n]);
else
N - Queens (k + 1, n);
}
}
```



Robot = 1
col = 1
Place (1, 1) safe
R1 → 1

Robot = 2
col = 1
Place (2, 1) ×
Place (2, 2) ×
place (2, 3) ✓
R2 → 3

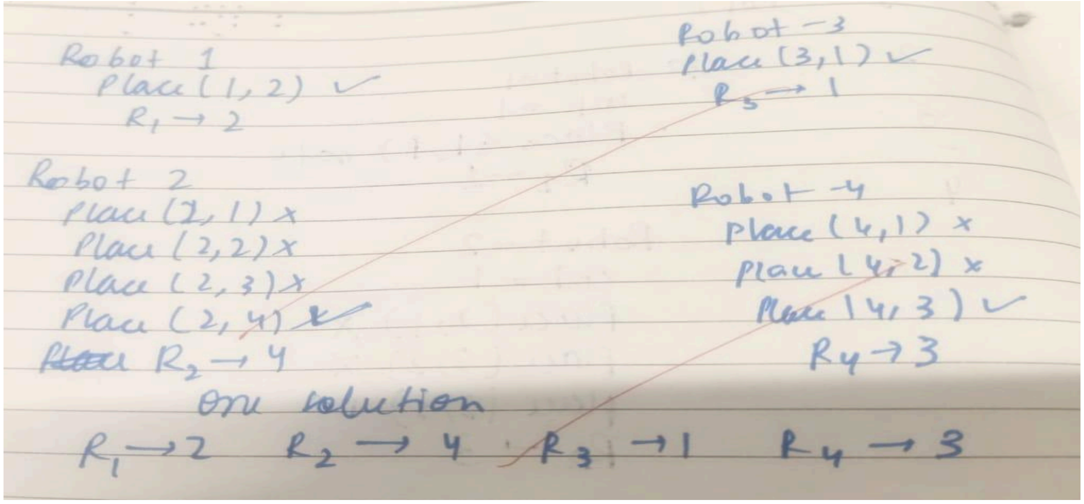
Robot = 3
Place (3, 1) ×
Place (3, 2) ×
Place (3, 3) ×
Place (3, 4) ×
Backtrack

Now
again Robot = 2
Place (2, 4) ✓
R2 → 4

Robot = 3
Place (3, 1) ×
Place (3, 2) ✓
R3 → 2

Robot = 4
Place (4, 1) ×
Place (4, 2) ×
Place (4, 3) ×
Place (4, 4) ×

Fail
Backtrack



Worst Case Time Complexity: O(N!)-1 mark

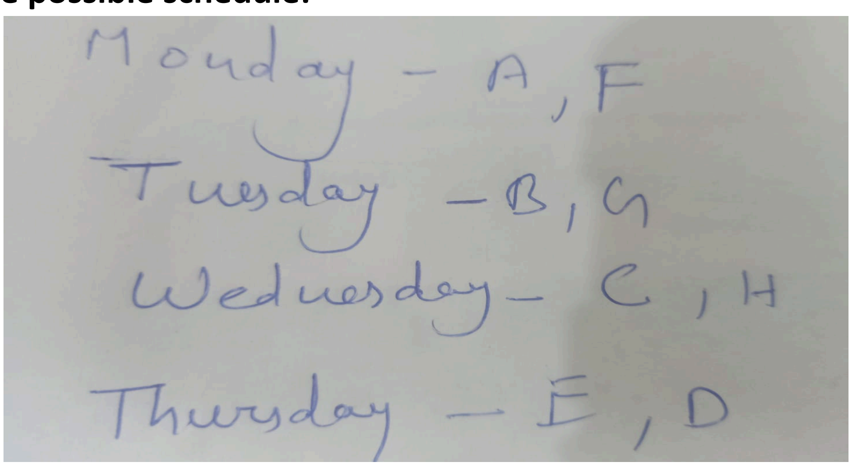
(b) TIET has a summer school session that offers eight courses (A-H). An "X" in **Table 2** shows which courses have students in common. There are time slots available (9:00 to 12:00 p.m.) each day (Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday) to schedule final exams. TIET wishes to schedule the eight final exams over as few days as possible without creating conflicts for the students scheduled to take them. Having more than one exam scheduled on the same day is fine as long as the courses do not have any students in common. Find the complete schedule of the exams. Show the steps involved in finding the schedule. Is it P or NP problem?

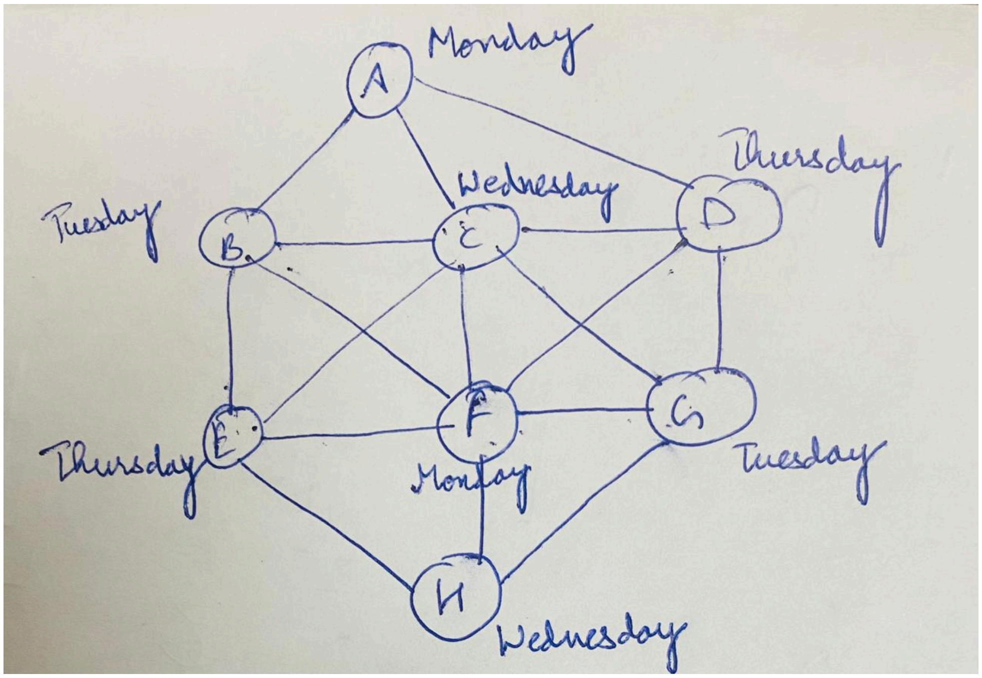
Table 2

	A	B	C	D	E	F	G	H
A		X	X	X				
B	X		X		X	X		
C	X	X		X	X	X	X	
D	X		X			X	X	
E		X	X			X		X
F		X	X	X	X		X	X
G			X	X		X		X
H					X	X	X	

Solution: complete-schedule with steps (4 marks)

one possible schedule:



	 **Chromatic number = 4 marks** **NP problem - 1 mark**	

Q.4	(a) Given a text "ABCCDDGCDD" along with a pattern "CDD", determine the location(s) of the pattern within the text by utilizing the Rabin Karp algorithm. Let d be the number of characters in the input set {A, B, C, .., J}. Further, choose q as 13 in the computation to ensure that all calculations can be carried out using single-precision arithmetic. Assign a numerical value to the characters in the input set {A, B, C, .., J} as {1, 2, 3, .., 10}. Show all the intermediate calculations.	(7)

Solution (4a)

Text "ABCCDDGCDD"

Pattern "CDD"

1) Hash value for pattern (p)

= ((3 * 10^2) + (4 * 10^1) + (4 * 10^0)) mod 13

= 344 mod 13

= 6

(8)

t0) Calculate the hash value for the text-window.

For the first window ABC,

Hash value for text(t) =

= ((1 * 10^2) + (2 * 10^1) + (3 * 10^0)) mod 13

= 123 mod 13 = 6

The hash value of the text pattern is compared to the hash value of the given text.

In the event of a match, character-matching is executed. The hash value of the initial window (t) corresponds to p, thus initiating character matching between ABC and CDD.

As they do not match, the next window is then considered.

t1) For the second window BCC,

Hash value for text(t)

= ((2 * 10^2) + (3 * 10^1) + (3 * 10^0)) mod 13

= 233 mod 13

= 12

We utilize the preceding hash value in the subsequent manner, for fast computation.

t = ((d * (t - v[character removed] * h) + v[character added]) mod 13

= ((10 * (6 - 1 * 9) + 3)mod 13

= 12

Where, h = 10^3-1 = 100 mod 13 = 9

For BCC, t = 12 ($\neq$6). Therefore, go for the next window.

t2) Next window CCD

Hash value for text(t) = Σ(v * dn-1) mod 13

= ((3 * 10^2) + (3 * 10^1) + (4 * 10^0)) mod 13

= 334 mod 13

= 9

For CCD, t = 9 (≠6). Therefore, go for the next window.

Next window CDD texthash = 6 = patternhash

Match Found (1st pattern found)

t3) Next window DDG

Hash value for text(t) = Σ(v * dn-1) mod 13

= ((4 * 10^2) + (4 * 10^1) + (7 * 10^0)) mod 13

= 447 mod 13

= 5

For DDG, t = 5 (≠6). Go for next window.

t4) Next window DGC

Hash value for text(t) = Σ(v * dn-1) mod 13

= ((4 * 10^2) + (7 * 10^1) + (3 * 10^0)) mod 13

= 473 mod 13

= 5

For DGC, t = 5 (≠6). Go for next window.

t5) Next window GCD

Hash value for text(t) = Σ(v * dn-1) mod 13

= ((7 * 10^2) + (3 * 10^1) + (4 * 10^0)) mod 13

= 734 mod 13

= 6

For GCD, t = 6 (= p). But pattern not matched.

t6) For next window, CDD

Next window CDD texthash = 6 = patternhash

Match Found (2nd pattern found)

Marking for 4a

Correct Hash for pattern = 0.5

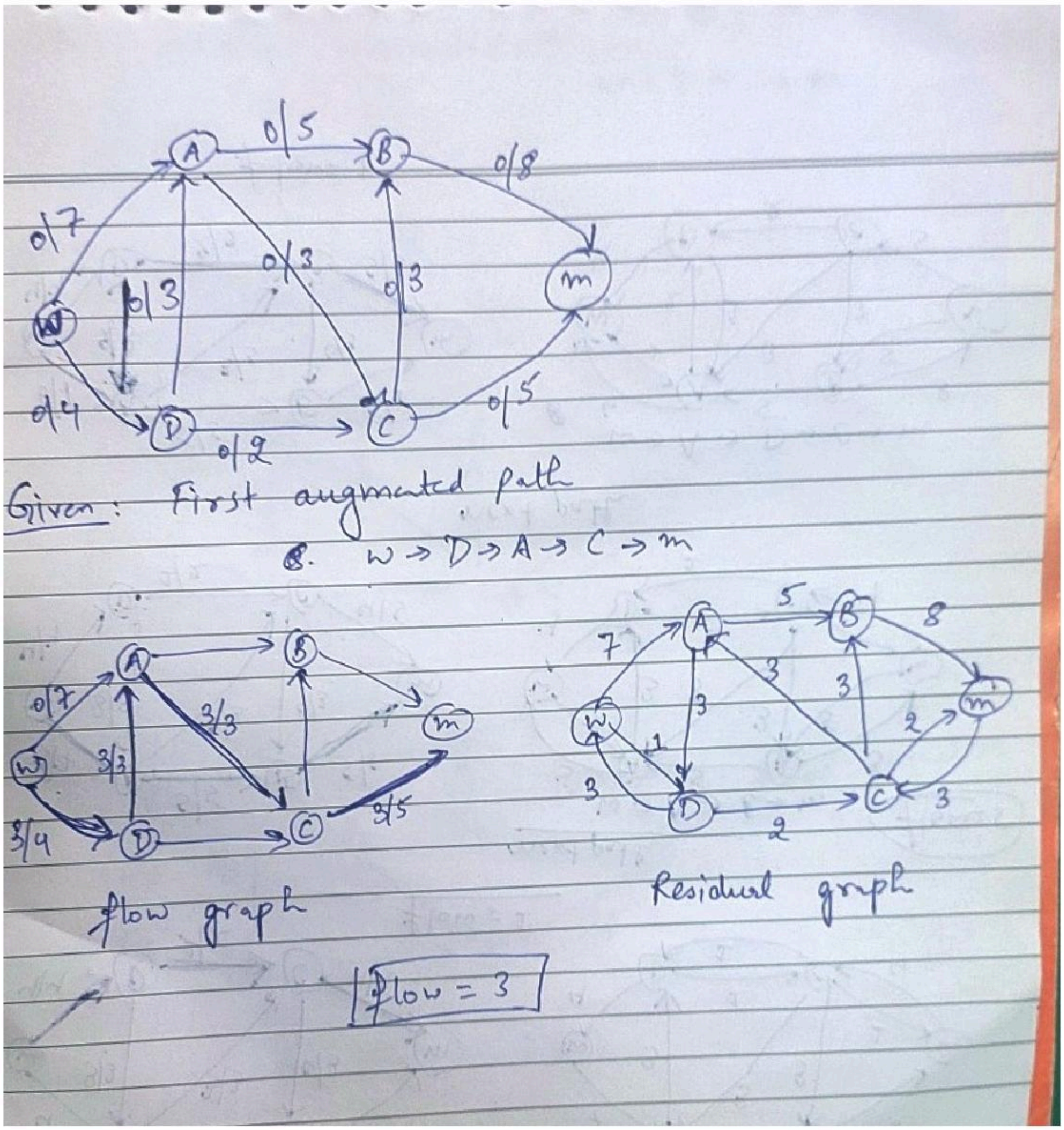
For each spurious hits = 1 *2 = 2

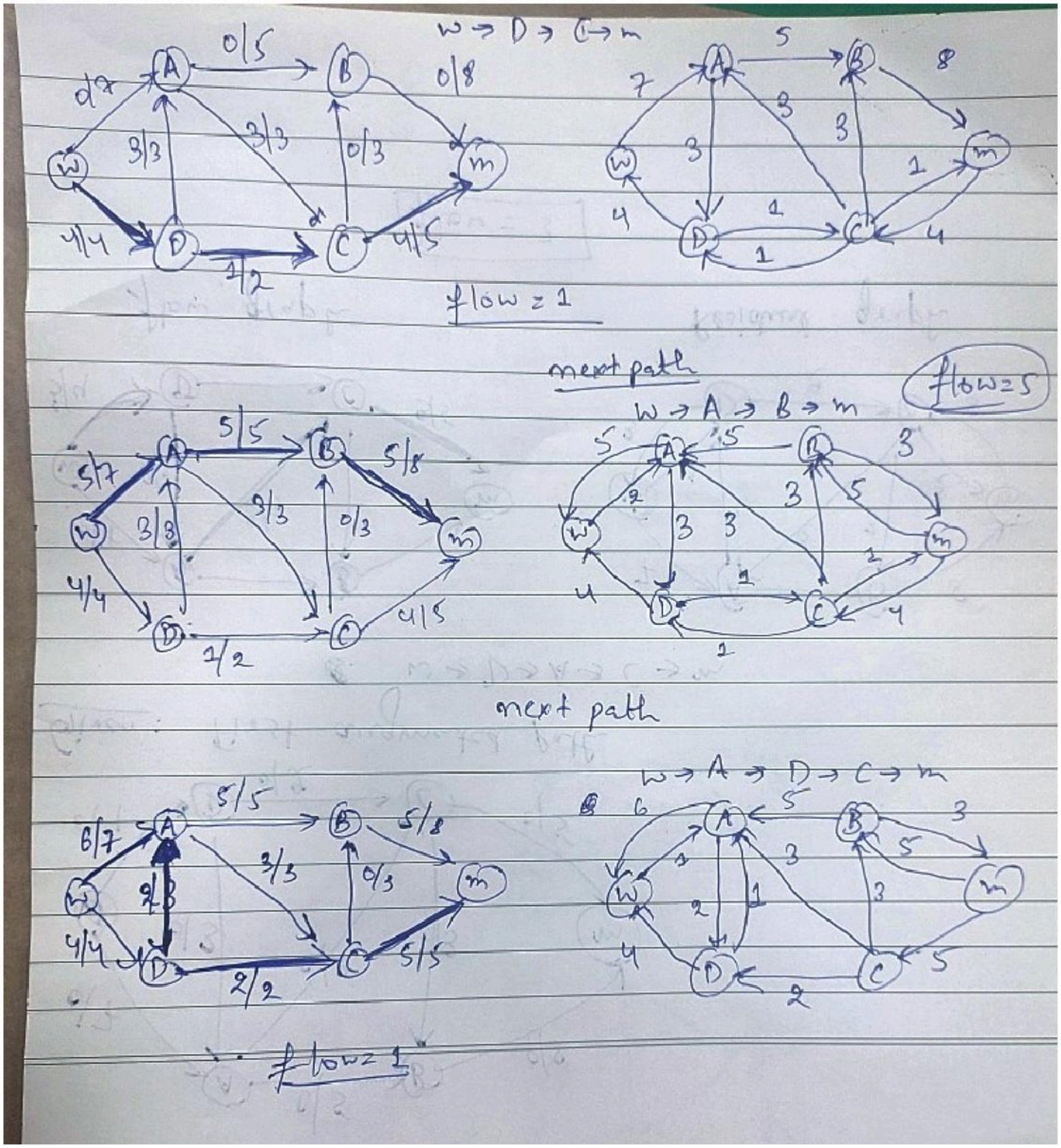
For each valid hits = 1*2 = 2

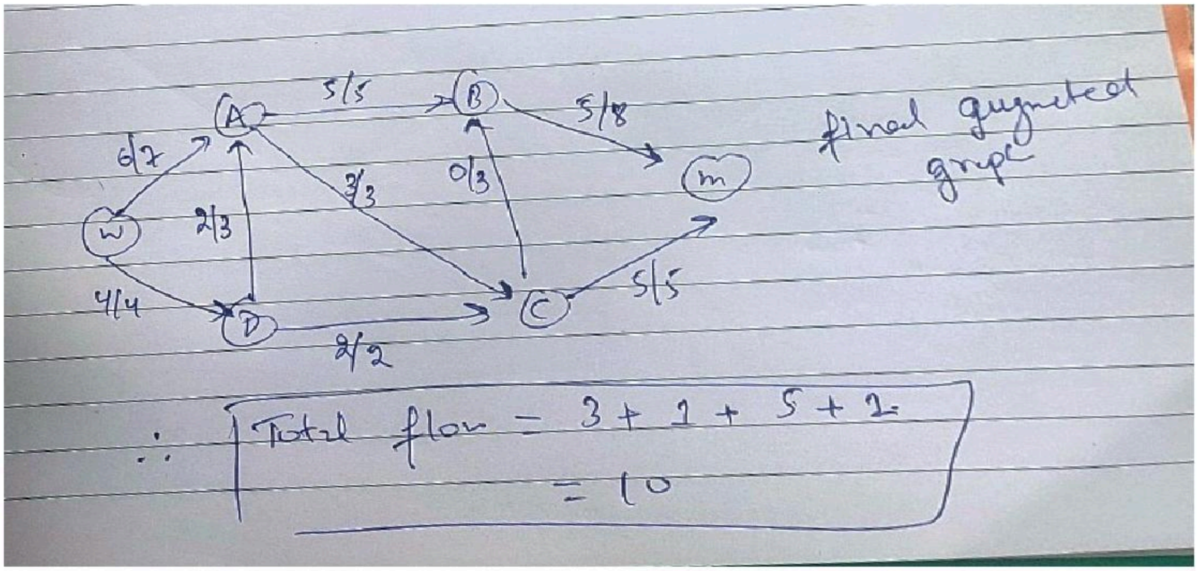
For each remaining correct text hash values = 0.5 *4=2

For correct index - 0.5

(b) Goods will be transported from a warehouse to 4 distribution centers: Ahmedabad, Bombay, Chennai, and Delhi, and finally to the market. The maximum number of goods that can be transferred from the warehouse to Ahmedabad is 7 units, and to Delhi is 4 units; from Ahmedabad to Bombay is 5 units; from Ahmedabad to Chennai is 3 units; from Delhi to Ahmedabad is 3 units; from Delhi to Chennai is 2 units; from Chennai to Bombay is 3 units; from Chennai to market is 5 units, and from Bombay to market is 8 units. This network configuration facilitates the efficient flow of goods from the central warehouse through the distribution centers to the final market destination. You need to find the maximum units of goods that can be transported from the warehouse to the market, showing all the intermediate stages of the residual graph. You need to select warehouse □ Delhi □ Ahmedabad □ Chennai □ market as the first augmented path in the flow network.




w → D → C → m
flow = 1
next path
w → A → B → m
flow = 5
next path
w → A → D → C → m
flow = 1



Marking (4b)

1.5 +2+2+2+0.5 = 8

Q.5 (10+3 +2)

Imagine you are tasked with surveying a set of 4 landmarks in a national park. A graph of these landmarks is shown in **Figure 1**, where the graph edges represent the distance between each landmark. You need to determine the most efficient route to visit each landmark exactly once and return to the starting point 'A.' You decided to apply two different algorithmic approaches:

i) Least Cost Branch-and-Bound (LCBB)

ii) 2-Approximation

You need to calculate the tour's total cost and path for both approaches by showing all the intermediate steps and state space tree for LCBB. Also, compare and contrast both LCBB and the 2-Approximation algorithms in terms of solution optimality and computational complexity (P, NP, NP-hard or NP-complete problems).

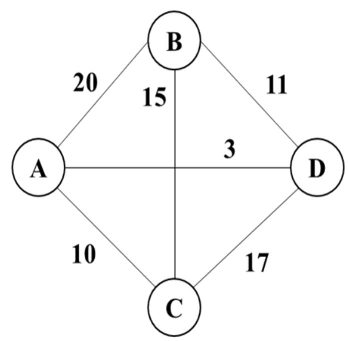
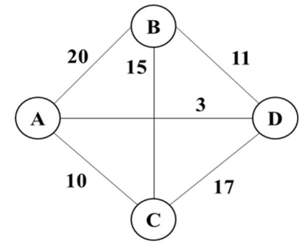


Figure 1

Solution:

i) LCBB

	A	B	C	D
A	∞	20	10	3
B	20	∞	15	11
C	10	15	∞	17
D	3	11	17	∞



	A	B	C	D	
A	∞	20	10	3	**3**
B	20	∞	15	11	**11**
C	10	15	∞	17	**10**
D	3	11	17	∞	**3**

	A	B	C	D
A	∞	17	7	0
B	9	∞	4	0
C	0	5	∞	7
D	0	8	14	∞
	0	**5**	**4**	**0**

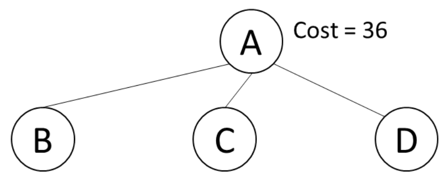
	A	B	C	D
A	∞	12	3	0
B	9	∞	0	0
C	0	0	∞	7
D	0	3	10	∞

Row Reduction cost = 3 + 11 + 10 + 3 = 27

Column Reduction cost = 0 + 5 + 4 + 0 = 9

Total reduction cost = 27 + 9 = 36

	A	B	C	D
A	∞	12	3	0
B	9	∞	0	0
C	0	0	∞	7
D	0	3	10	∞



	A	B	C	D
A	∞	∞	∞	∞
B	∞	∞	0	0
C	0	∞	∞	7
D	0	∞	10	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	∞	∞	0	0	**0**
C	0	∞	∞	7	**0**
D	0	∞	10	∞	**0**

No row reduction

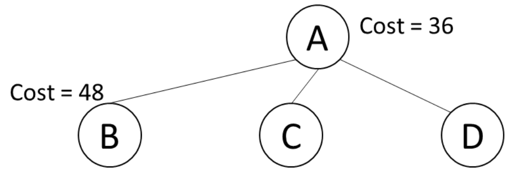
	A	B	C	D
A	∞	∞	∞	∞
B	∞	∞	0	0
C	0	∞	∞	7
D	0	∞	10	∞
	0		**0**	**0**

No column reduction

new reduction cost = 0

Total cost = Cost(AB) + new reduction + Cost(A)
= 12 + 0 + 36 = 48

	A	B	C	D
A	∞	12	3	0
B	9	∞	0	0
C	0	0	∞	7
D	0	3	10	∞



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	0	∞	7
D	0	3	∞	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	9	∞	∞	0	0
C	∞	0	∞	7	0
D	0	3	∞	∞	0

No row reduction

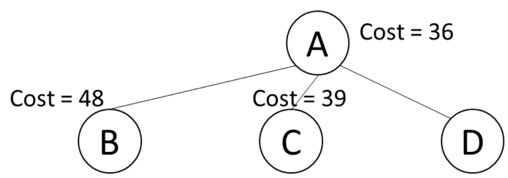
	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	0	∞	7
D	0	3	∞	∞
	0	0		0

No column reduction

new reduction cost = 0

Total cost = Cost(AC) + new reduction + Cost(A)
= 3 + 0 + 36 = 39

	A	B	C	D
A	∞	12	3	0
B	9	∞	0	0
C	0	0	∞	7
D	0	3	10	∞



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	0	∞
C	0	0	∞	∞
D	∞	3	10	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	9	∞	0	∞	0
C	0	0	∞	∞	0
D	∞	0	10	∞	3

Row reduction cost = 3

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	0	∞
C	0	0	∞	∞
D	∞	0	7	∞
	0	0	0	

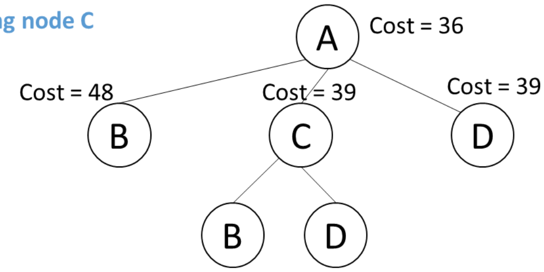
No column reduction

new reduction cost = 3

Total cost = Cost(AD) + new reduction + Cost(A)
= 0 + 3 + 36 = 39

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	0	∞	7
D	0	3	∞	∞

Case 1: Choosing node C



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	∞	∞	∞
D	0	∞	∞	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	9	∞	∞	0	**0**
C	∞	∞	∞	∞	
D	0	∞	∞	∞	**0**

No row reduction

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	∞	∞	∞
D	0	∞	∞	∞
	0			**0**

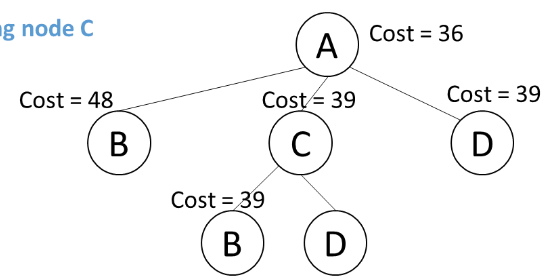
No column reduction

new reduction cost = 0

Total cost = Cost(CB) + new reduction + Cost(C)
= 0 + 0 + 39 = 39

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	0	∞	7
D	0	3	∞	∞

Case 1: Choosing node C



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	∞
C	∞	∞	∞	∞
D	0	3	∞	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	0	∞	∞	∞	**9**
C	∞	∞	∞	∞	
D	0	3	∞	∞	**0**

Row reduction cost = 9

	A	B	C	D
A	∞	∞	∞	∞
B	0	∞	∞	∞
C	∞	∞	∞	∞
D	0	0	∞	∞
	0	**3**		

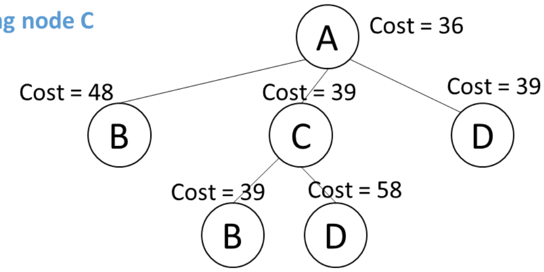
Column reduction cost = 3

new reduction cost = 12

Total cost = Cost(CD) + new reduction + Cost(C)
= 7 + 12 + 39 = 58

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	0
C	∞	0	∞	7
D	0	3	∞	∞

Case 1: Choosing node C



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	∞
C	∞	∞	∞	∞
D	0	3	∞	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	0	∞	∞	∞	9
C	∞	∞	∞	∞	
D	0	3	∞	∞	0

Row reduction cost = 9

	A	B	C	D
A	∞	∞	∞	∞
B	0	∞	∞	∞
C	∞	∞	∞	∞
D	0	0	∞	∞
	0	3		

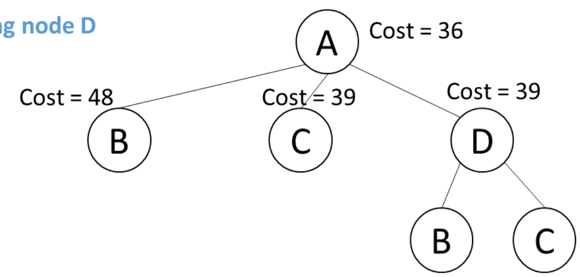
Column reduction cost = 3

new reduction cost = 12

Total cost = Cost(CD) + new reduction + Cost(C)
= 7 + 12 + 39 = 58

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	0	∞
C	0	0	∞	∞
D	∞	0	7	∞

Case 2: Choosing node D



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	0	∞
C	0	∞	∞	∞
D	∞	∞	∞	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	9	∞	0	∞	0
C	0	∞	∞	∞	0
D	∞	∞	∞	∞	

No row reduction

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	0	∞
C	0	∞	∞	∞
D	∞	∞	∞	∞
	0		0	

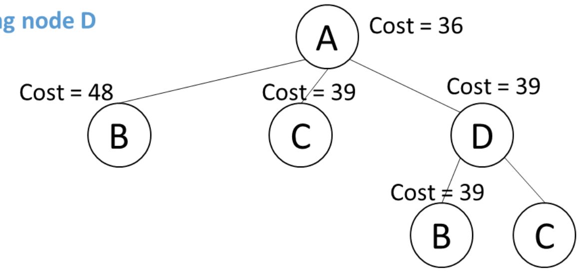
No column reduction

new reduction cost = 0

Total cost = Cost(DB) + new reduction + Cost(D)
= 0 + 0 + 39 = 39

	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	0	∞
C	0	0	∞	∞
D	∞	0	7	∞

Case 2: Choosing node D



	A	B	C	D
A	∞	∞	∞	∞
B	9	∞	∞	∞
C	0	0	∞	∞
D	∞	∞	∞	∞

	A	B	C	D	
A	∞	∞	∞	∞	
B	0	∞	∞	∞	9
C	0	0	∞	∞	0
D	∞	∞	∞	∞	

Row reduction cost =9

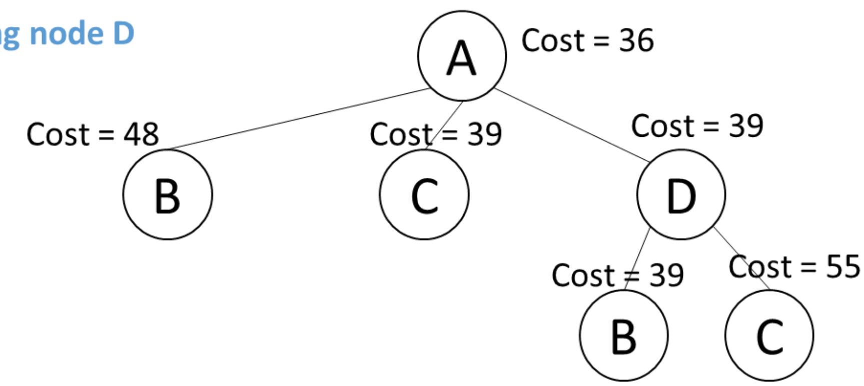
	A	B	C	D
A	∞	∞	∞	∞
B	0	∞	∞	∞
C	0	0	∞	∞
D	∞	∞	∞	∞
	0	0		

No column reduction

new reduction cost = 9

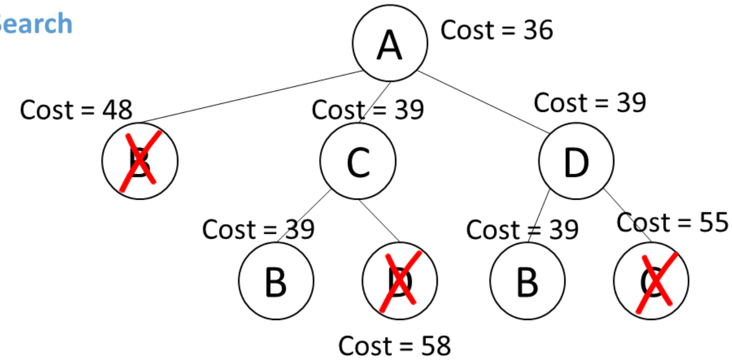
Total cost = Cost(DC) + new reduction + Cost(D)
= 7 + 9 + 39 = 55

Case 2: Choosing node D



Tour = $A \rightarrow D \rightarrow B \rightarrow C \rightarrow A$
Length = 39

Case 3: Exhaustive Search

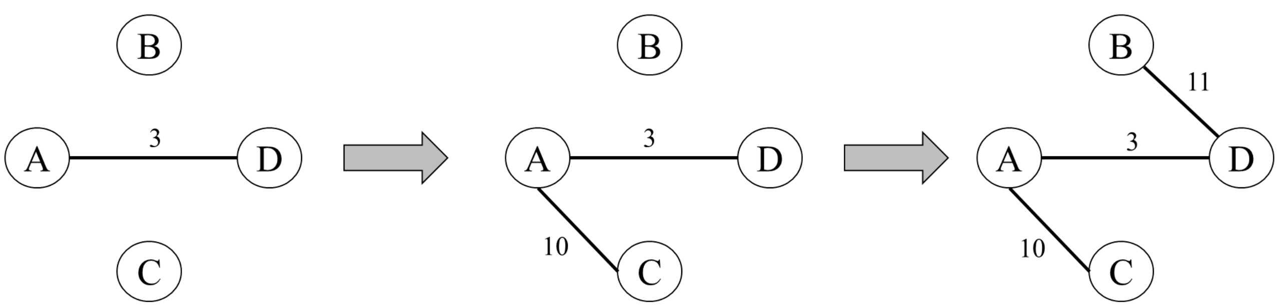


Either
Tour = $A \rightarrow C \rightarrow B \rightarrow D \rightarrow A$
Length = 39

Or
Tour = $A \rightarrow D \rightarrow B \rightarrow C \rightarrow A$
Length = 39

6 Marks for 6 Matrices
2 Marks for the state space diagram
1 Mark for the final tour
1 Mark for tour length

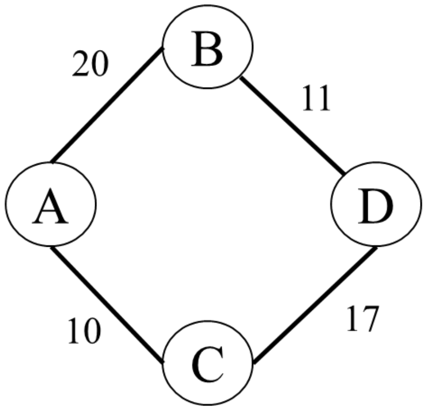
ii) 2- approximation



Prim's MST

Either

Preorder: A C D B

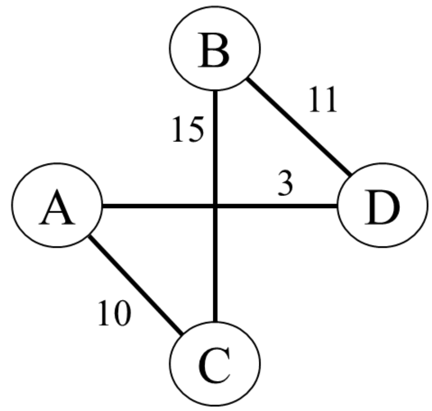


Tour = $A \to C \to D \to B \to A$

Length = 58

Or

Preorder: A D B C



Tour = $A \to D \to B \to C \to A$

Length = 39

1 Mark for the Prim's MST
1 Mark for the preorder traversal
0.5 Mark for tour
0.5 Mark for tour length

Difference:
1 Mark for the Optimality
1 Mark for the P or NP