\

Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2025-26

Class:	T.E.	**Semester:**	V
Course Code:	CSC501	**Course Name:**	COMPUTER NETWORKS

Name of Student:	SHRUTI GAUCHANDRA
Roll No. :	18
Assignment No.:	03
Title of Assignment:	SUBNETTING, ROUTING ALGORITHMS
Date of Submission:	~~11~~ 08/25
Date of Correction:	11/08/25

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Completeness	5	05
Demonstrated Knowledge	3	03
Legibility	2	02
Total	10	10

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Completeness	5	3-4	1-2
Demonstrated Knowledge Legibility	3	2	1
Legibility	2	1	0

Checked by

Name of Faculty : Mrs. SNEHA YADAV

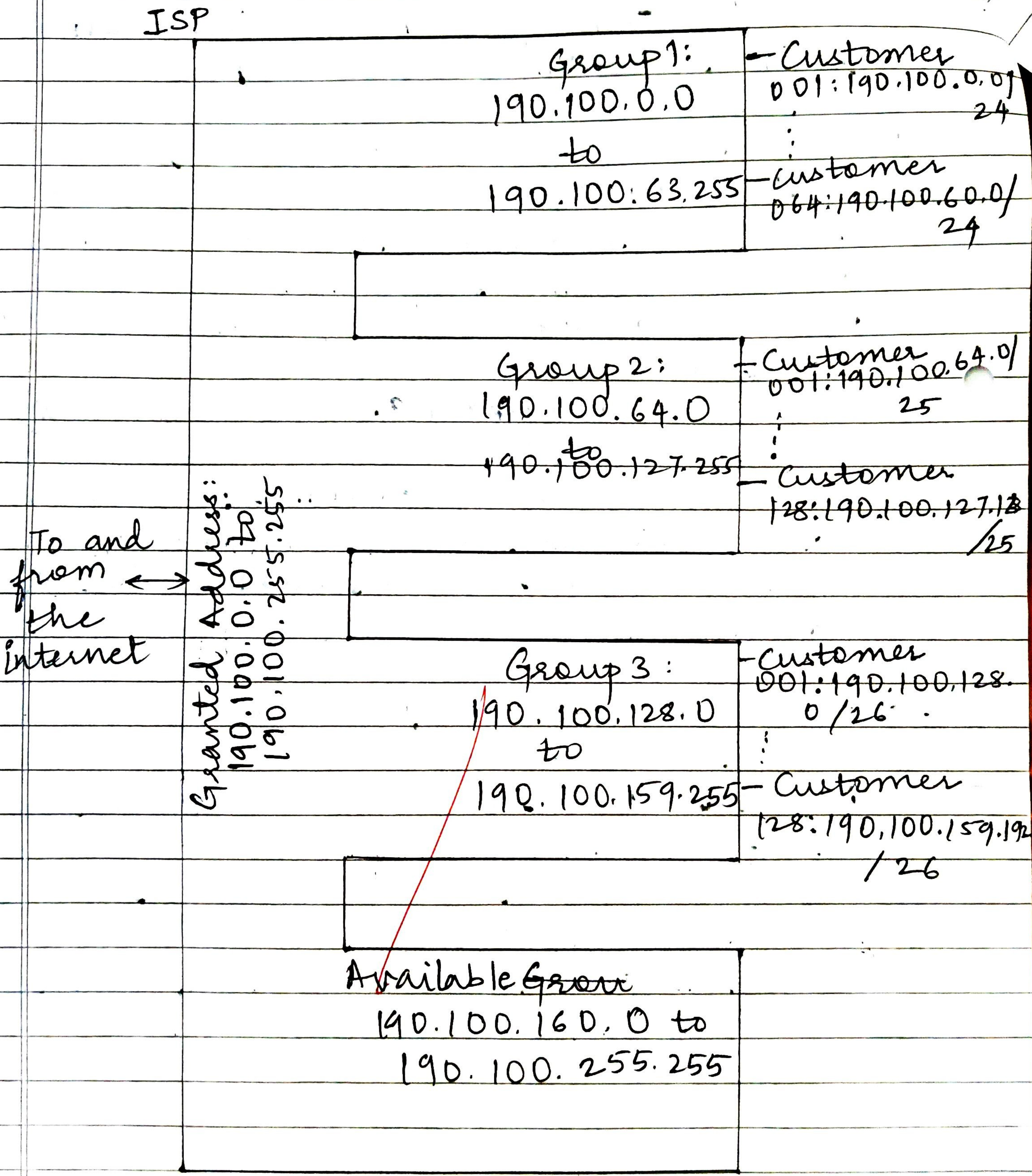
Signature :

Date : 11/08

CSC503.3 Apply Subnetting, Network Address Translation and routing algorithms for shortest paths.

Q1. An ISP is granted a block of address starting with 190.100.0.0/16 (65,536 addresses).
The ISP needs to distribute these addresses to three groups of customers as follows:
1. The first group has 64 customers; each needs 256 addresses.
2. The second group has 128 customers; each needs 128 addresses.
3. The third group has 128 customers; each needs 64 addresses. Design the subblocks and find out how many addresses are still available after these allocations.

→ ISP needs to distribute these addresses to three groups of customers as follows:
1) The first group has 64 customers, each needs 256 addresses.
2) The second group has 128 customers, each needs 128 addresses.
3) The third group has 128 customers, each needs 64 addresses.



An address allocation and distribution by an ISP.

1) Group 1 : The first group has 64 customers, each needs 256 addresses.

In this group, each customer require 256 addresses. This means that 8 ($\log_2 256$) bits are needed to define each host. The prefix length is then $32-8=24$.

The addresses are:

1st customer : 190.100.0.0/24

⋮

190.100.0.255/24

2nd customer : 190.100.1.0/24

190.100.1.255/24

⋮

64th customer : 190.100.63.0/24

190.100.63.255/24

Total $= 64 \times 256$

$= 16384$

2) Group 2 : The second group has 128 customers, each needs 128 addresses.

In this group, each customer needs 128 addresses. This means that 7 ($\log_2 128$) bits are needed to define each host.

The prefix length is $32-7=25$

The addresses are:

1^{st} customer: 190.100.64.0/25

190.100.64.127/25

2^{nd} customer: 190.100.64.128/25

190.100.64.255/25

⋮

128^{th} customer: 190.100.127.128/25

190.100.127.255/25

Total = 128 × 128
= 16384.

3) Group 3: The third group has 128 customers, each needs 64 addresses.

In this group, each customer needs 64 addresses. This means that 6 ($\log_2 64$) bits are needed to each host.

Prefix length is 32 − 6 = 26

The addresses are:

1^{st} customer: 190.100.128.0/26

190.100.128.63/26

2^{nd} customer: 190.100.128.64/26

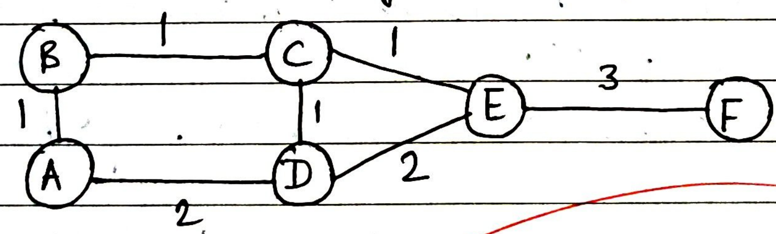
190.100.128.127/26

3rd customer : 190.100.159.192/26
190.100.159.255/26

Total = 128 X 64
= 8192

Number of granted addresses to ISP = 65536
Number of allocated addresses by ISP = 40960
Number of available addresses = 24576

Q2. For the network shown in fig., show the computations at node A using the Djkstra's Algorithm.



→ Step 1:

(1) Since computation are to be done at node A, starting node will be A.
We enter this node into group P.

(2) We add neighbouring node B and D in group T along with costs to reach them through A.

Permanent (P)	Temporary (T)
A	B(A, 1), D(A, 2)

A

Step 2:

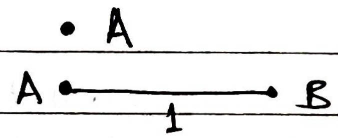
① Now pick up neighbour with smallest cost and add it to P set.
Here neighbour with smallest cost is B.
So let's add B(A,1) to P group.

② As B is added to P group, we have to add it's neighbour i.e C to T group.

Permanent (P)	Temporary (T)
A	B(A,1), D(A,2)
A, B(A,1)	D(A,2), C(B,2)

1

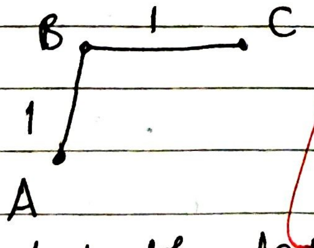
2



③ Note that D(A,2) has remained in T group as it is but C(B,2) is new entry.
C(B,2) means C is reached by A via B with a cost 2

④ The cost is 2 due to path followed from A to B and then to C

3



cost of path followed from A to C is
: 1+1 = 2

Step 3:

(1) Now pick up neighbour in T set with smallest cost and add it to P set. Here we choose neighbour D because it's immediate neighbour of A.

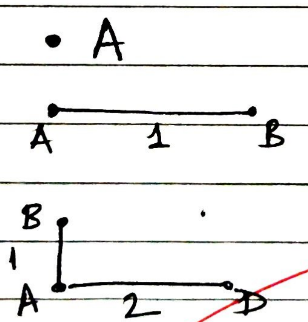
(2) Since D is added to P group, we have to add it's neighbour i.e. C and E to T group.
Note that C(B,2) goes as it is and E(D,4) is new entry to P.
But C(D,3) can't be entered because it's cost is 3.

Permanent (P)	Temporary (T)
A	B(A,1), D(A,2)
A, B(A,1)	D(A,2), C(B,2)
A, B(A,1), D(A,2)	E(D,4), C(B,2)

1.

2.

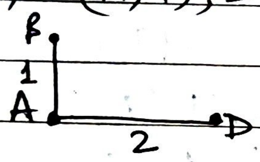
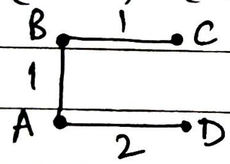
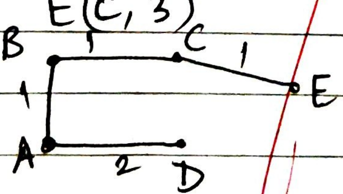
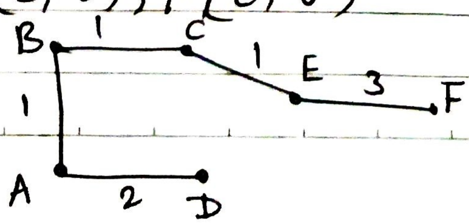
3.



where E(D,4) means E is reached by A via D and cost is 4.
Similarly we can proceed further.

FOR EDUCATIONAL USE

Final table:

Permanent (P)	Temporary (T)
A • A	B(A,1), D(A,2)
A, B(A,1) 	D(A,2), C(B,2)
A, B(A,1), D(A,2) 	E(D,4), C(B,2)
A, B(A,1), D(A,2), C(B,2) 	E(C,3), E(D,4) can't be included
A, B(A,1), D(A,2), C(B,2), E(C,3) 	F(E,6), F(E,7) can't be included
A, B(A,1), D(A,2), C(B,2), E(C,3), F(E,6) 	Empty NULL

Shortest paths from A to all other nodes