	PUNE INSTITUTE OF COMPUTER TECHNOLOGY PUNE - 411043	
	Department of Electronics & Telecommunication	
	ASSESMENT YEAR: 2020-2021	CLASS: SE 5
	SUBJECT: DATA STRUCTURES	
EXPT No: 10	LAB Ref: SE/2020-21/	Starting date: 04/12/2020
	Roll No: 22119	Submission date: 04/12/2020
Title:	Prim's Algorithm	

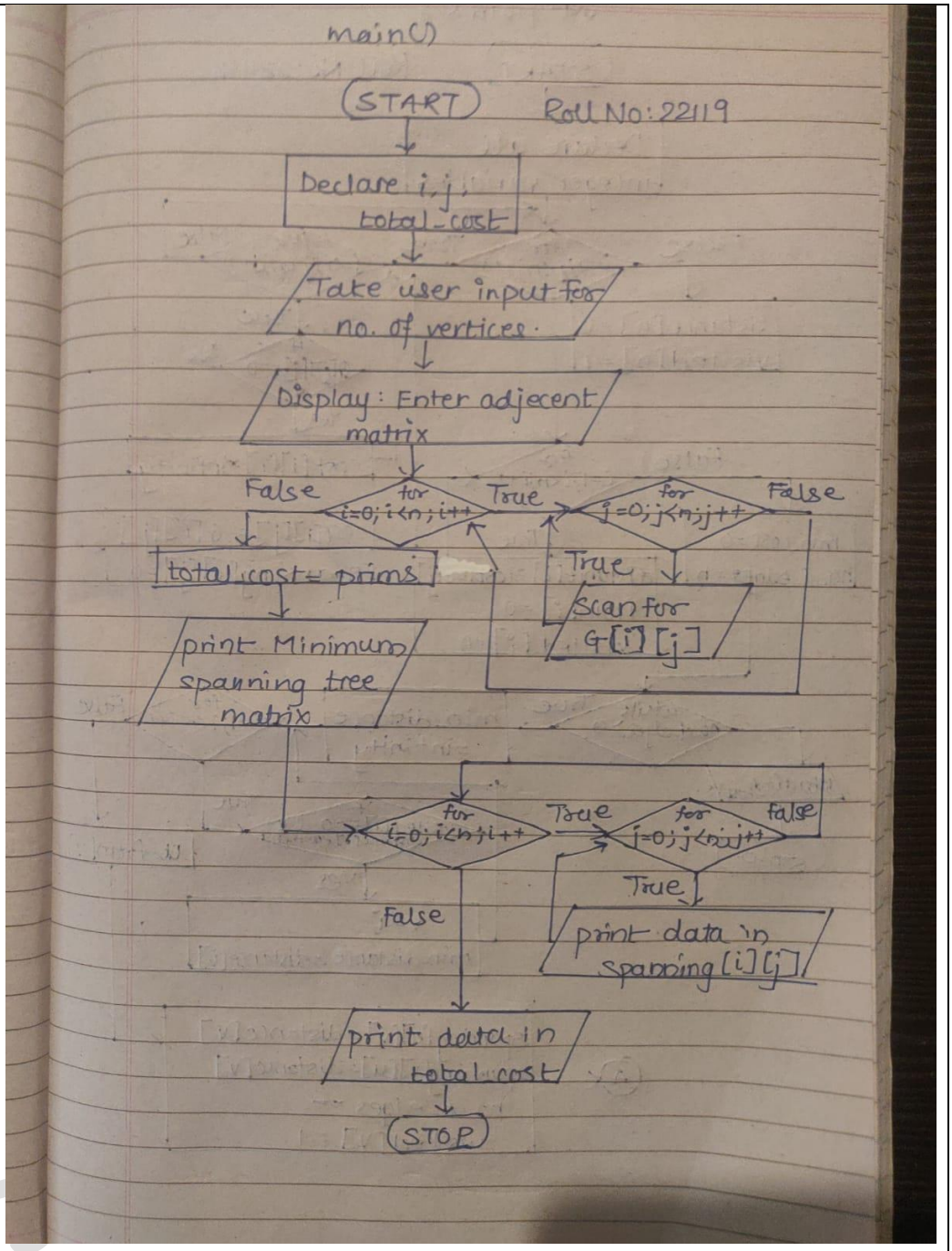
Prerequisites:	• DEVC++ IDE • Knowledge about Prims Algorithm and its working • Knowledge about cyclic data structures
Objectives:	• To learn the concepts of graph (cyclic data structure). • Generate spanning tree using Prim's Algorithm.

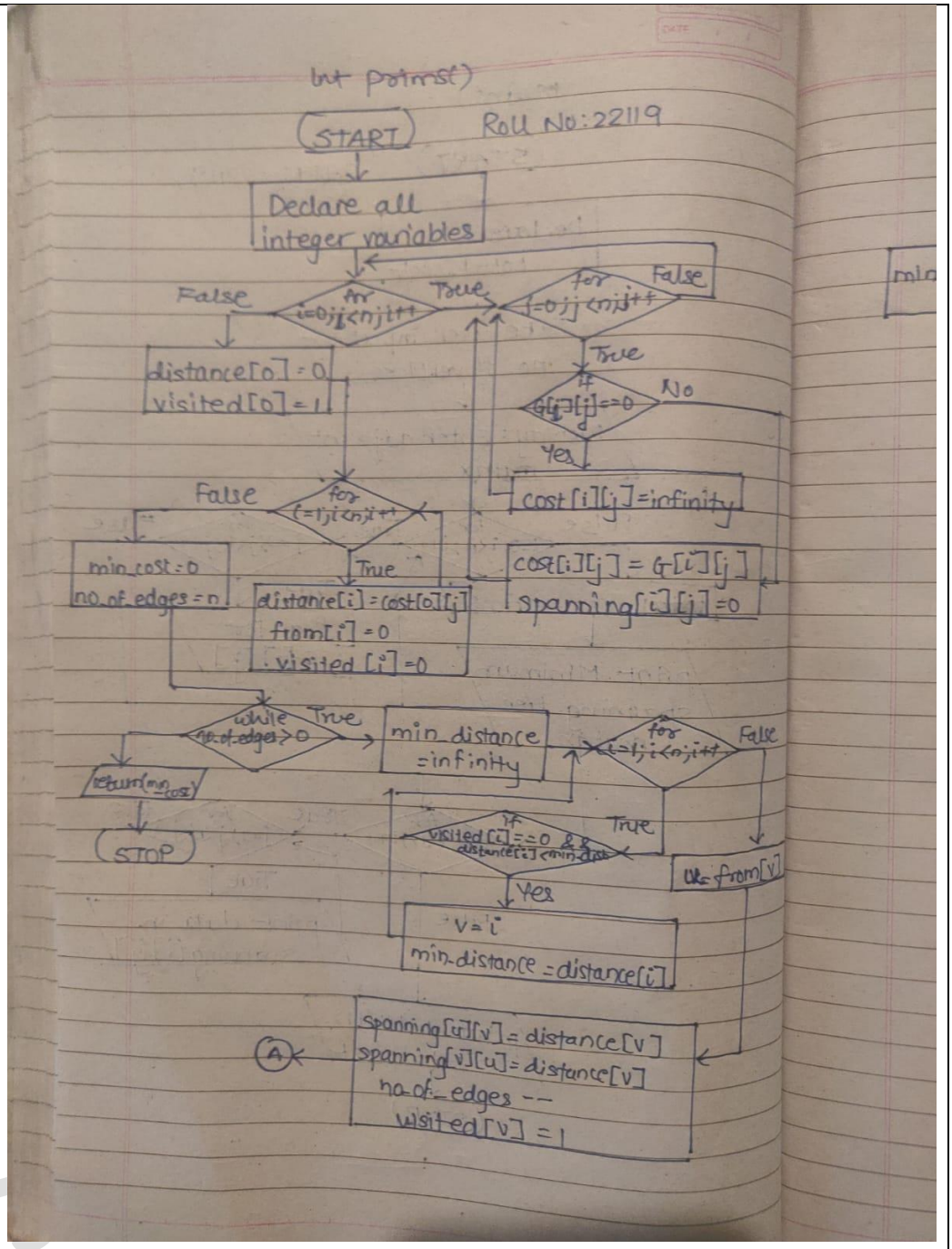
Theory:

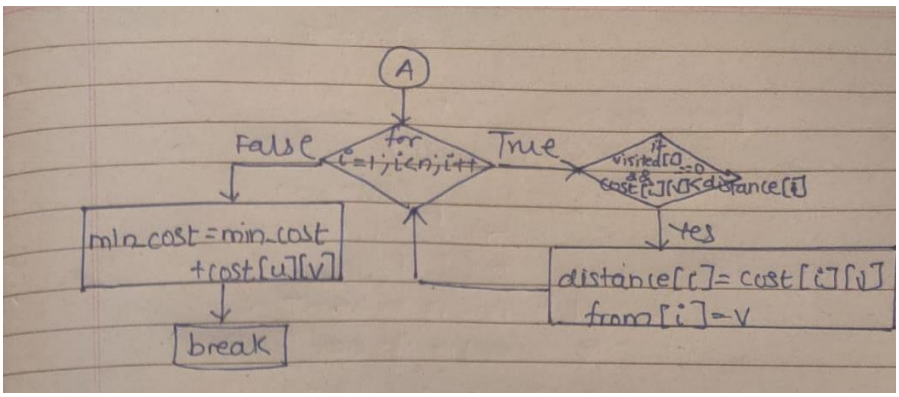
Prim's Algorithm is an approach to determine minimum cost spanning tree. In this case, we start with single edge of graph and we add edges to it and finally we get minimum cost tree. In this case, as well, we have n-1 edges when number of nodes in graph are n. We again and again add edges to tree and tree is extended to create spanning tree

Algorithm	This algorithm creates spanning tree with minimum weight from a given weighted graph. 1) Begin 2) Create edge list of given graph, with their weights. 3) Draw all nodes to create skeleton for spanning tree. 4) Select an edge with lowest weight and add it to skeleton and delete edge from edge list. 5) Add other edges. While adding an edge take care that the one end of the edge should always be in the skeleton tree and its cost should be minimum. 6) Repeat step 5 until n-1 edges are added. 7) Return.

Flow-chart


main()
START
Roll No: 22119
Declare i, j, total_cost
Take user input for no. of vertices.
Display: Enter adjecent matrix
for i=0; i<n; i++
False
True
for j=0; j<n; j++
False
True
Scan for G[i][j]
total_cost = prims
print Minimum spanning tree matrix
for i=0; i<n; i++
True
for j=0; j<n; j++
false
True
print data in spanning[i][j]
False
print data in total_cost
STOP


int prims()
START
Roll No: 22119
Declare all integer variables
for i=0;i<n;i++
False
True
distance[0] = 0
visited[0] = 1
for i=0;j<n;j++
True
False
if G[i][j]==0
No
Yes
cost[i][j]=infinity
cost[i][j] = G[i][j]
spanning[i][j]=0
for i=1;i<n;i++
False
True
min_cost=0
no_of_edges=n
distance[i]=cost[0][j]
from[i]=0
visited[i]=0
while no_of_edges>0
True
min_distance =infinity
for i=1;i<n;i++
False
return(min_cost)
STOP
if visited[i]==0 && distance[i]<min_dist
True
u=from[v]
Yes
v=i
min_distance=distance[i]
A
spanning[u][v]=distance[v]
spanning[v][u]=distance[v]
no_of_edges--
visited[v]=1


A
for i=1;i<n;i++
False
True
if visited[i]==0 && cost[u][i]<distance[i]
Yes
min_cost=min_cost +cost[u][v]
break
distance[i]=cost[i][v]
from[i]=v

ERROR	None
REMEDY	None
CONCLUSION:	
	• Hence, we have generated the minimum spanning tree using the Prims algorithm • Concept of graphs has ben understood
REFERENCES:	
	1) Seymour Lipschutz, Data Structure with C, Schaum's Outlines, Tata McGrawHill 2) Yedidyah Langsam – Data structures using C and C++ - PHI Publications (2nd Edition). 3) Yashavant Kanetkar, Data Structures Through C, BPB Publication, 2nd Edition

Continuous Assessment			**Assessed By**
RPP (5)	**ARR (5)**	**Total (10)**	**Signature:**
			Date: