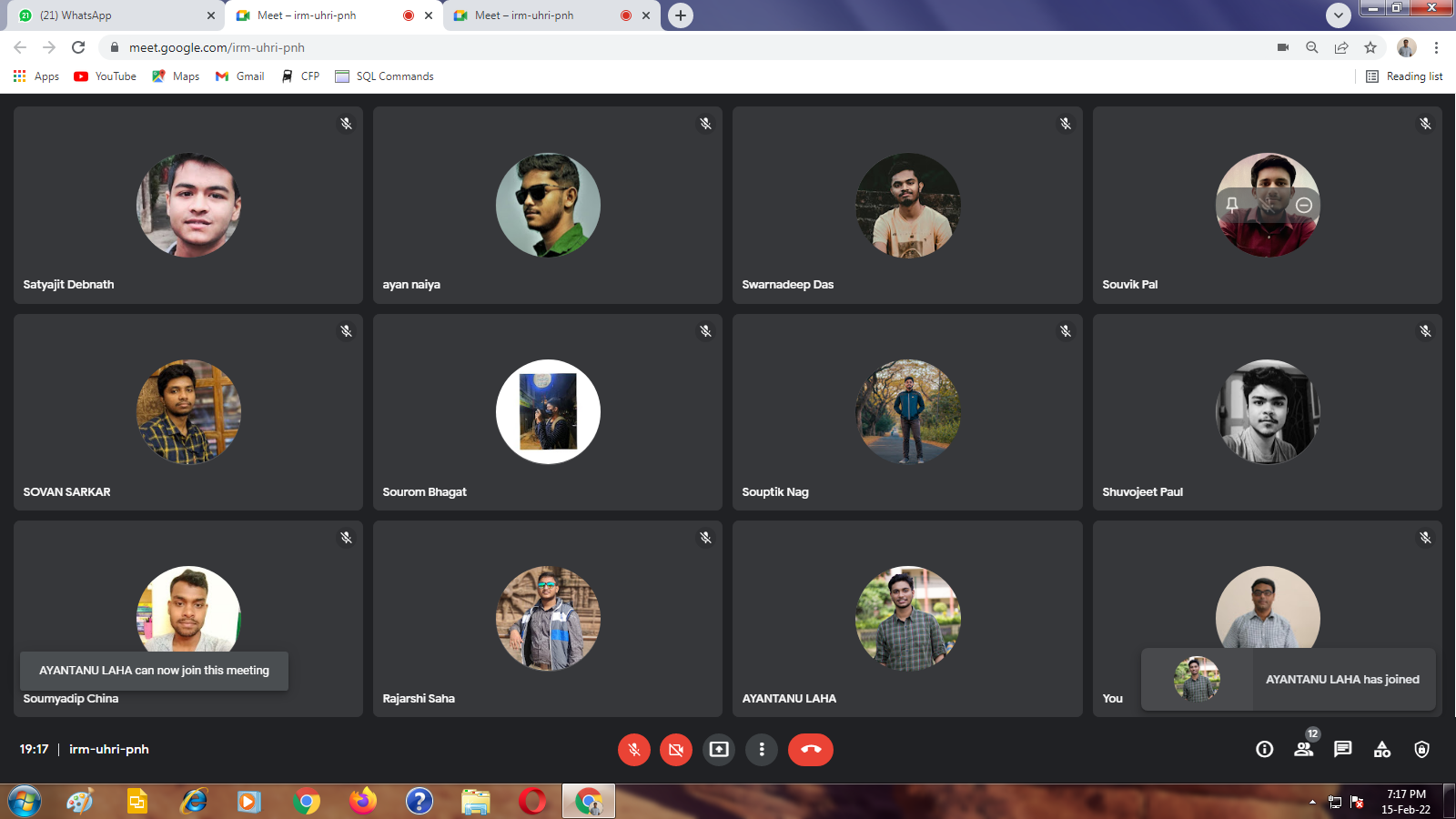
**CLASS 15/02/2022**

**UG SEM3**

**GRAPH THEORY**



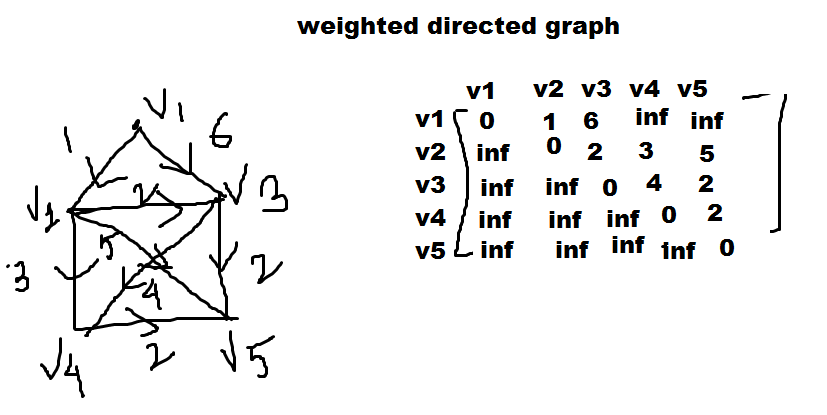
**SHORTEST PATH ALGORITHM**

1. One source to all destination vertices.
2. One source to one destination
3. All source to all desination

* Dijkstra algorithm
* Floyd warshall algorithm
* Warshall algorithm

**DIJKSTRA ALGORITHM**

* It is an optimization algorithm which finds the shortest path from a source vertex to all other vertices/destination vertex of a directed weighted graph.

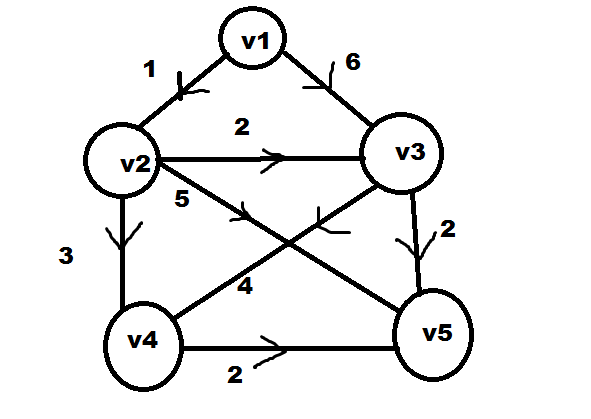


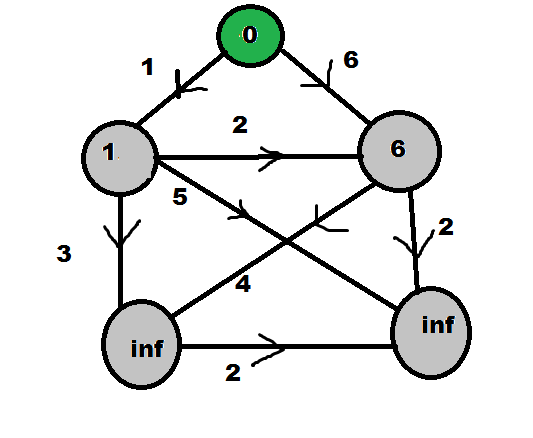
A weighted directed graph G of n vertices can be described by a matrix D=[dij]

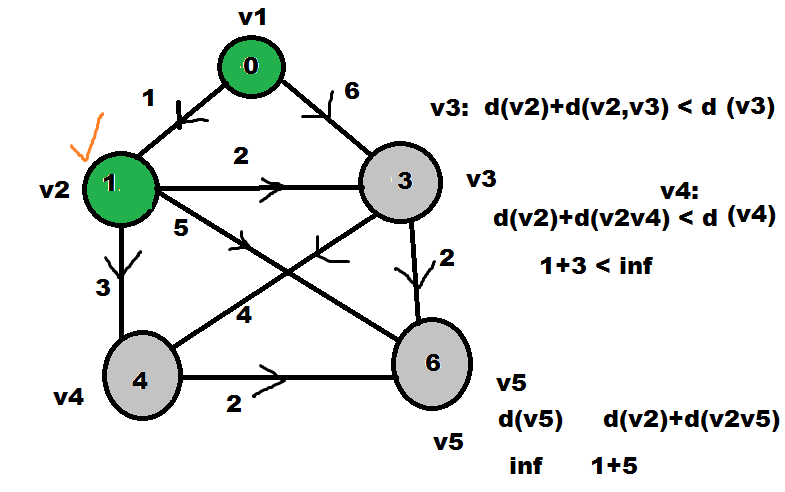
d(i,j)= weight of directed edge from vertex i to j

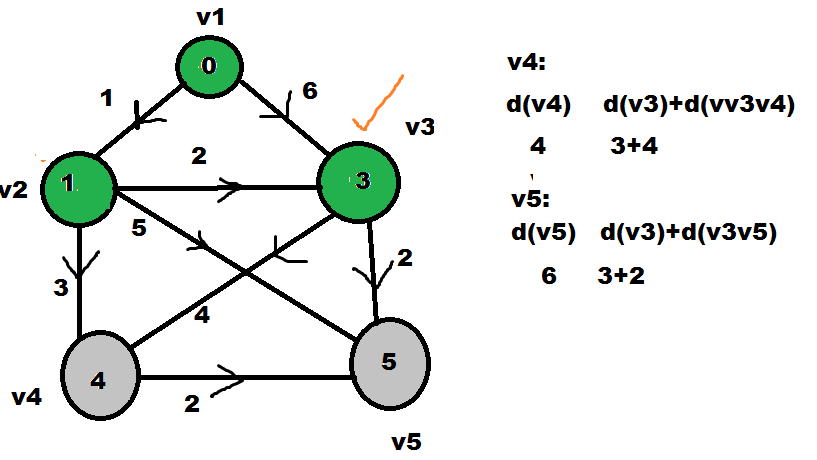
d(i,i)=0

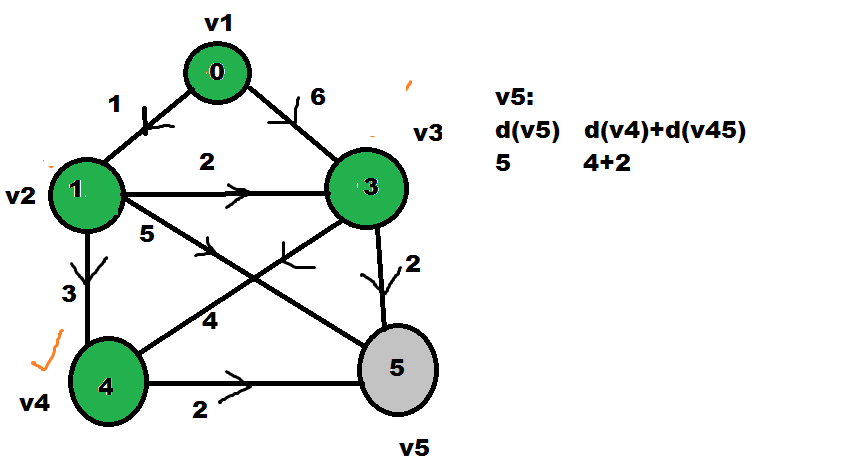
d(i,j)=infinity if there is no edge form i to j

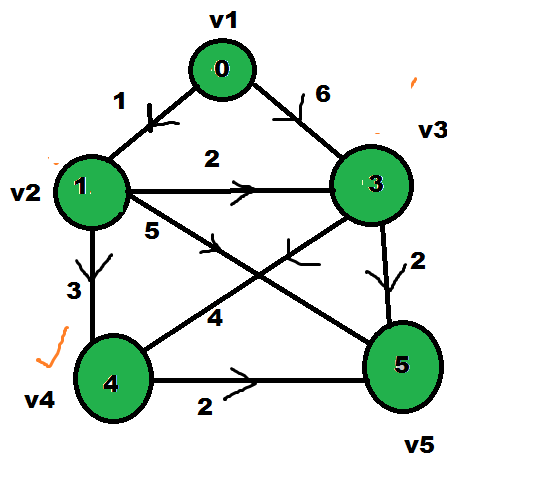


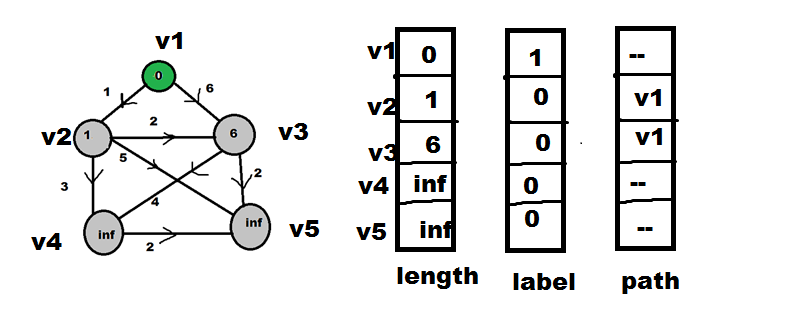




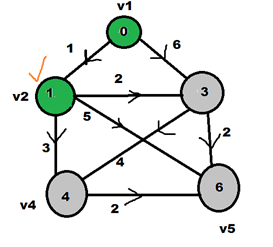




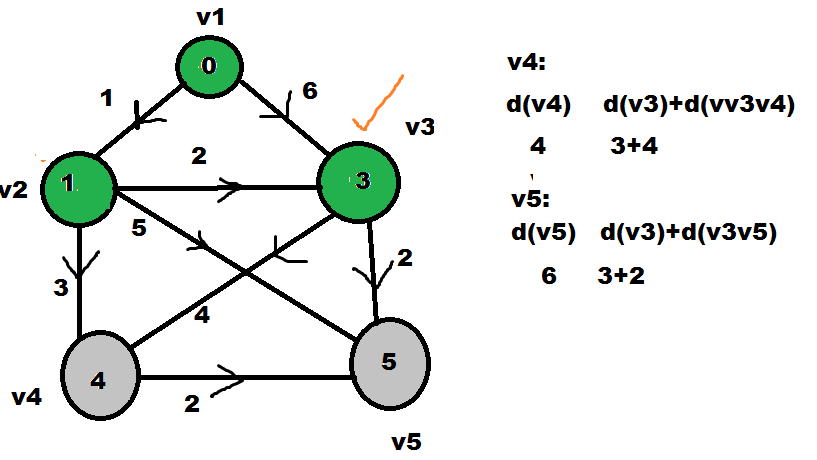




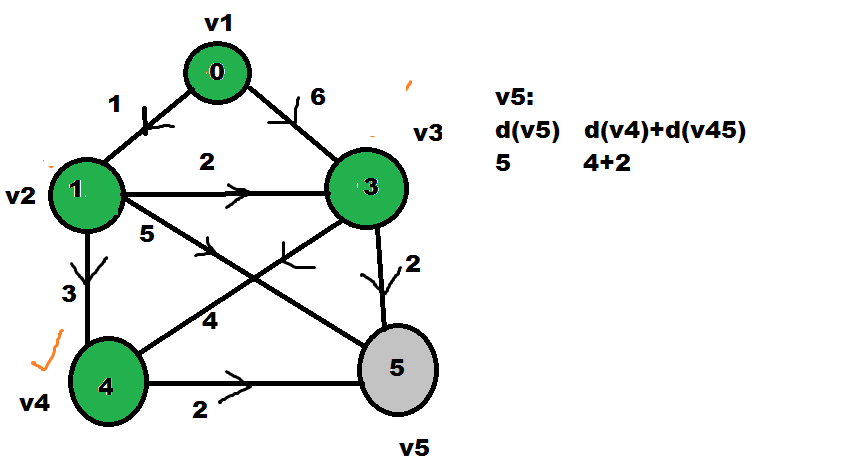
|  |  |  |
| --- | --- | --- |
| **length** | **label** | **path** |
| **0** | **1** | **--** |
| **1** | **1** | **v1** |
| **3** | **0** | **v2** |
| **4** | **0** | **v2** |
| **6** | **0** | **v2** |



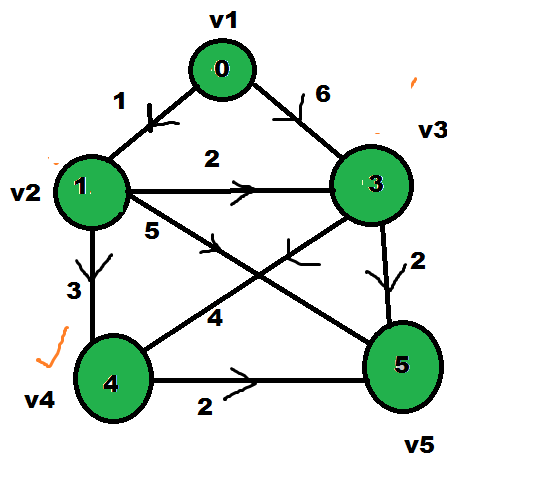
|  |  |  |
| --- | --- | --- |
| **length** | **label** | **path** |
| **0** | **1** | **--** |
| **1** | **1** | **v1** |
| **3** | **1** | **v2** |
| **4** | **0** | **v2** |
| **5** | **0** | **v3** |



|  |  |  |
| --- | --- | --- |
| **length** | **label** | **path** |
| **0** | **1** | **--** |
| **1** | **1** | **v1** |
| **3** | **1** | **v2** |
| **4** | **1** | **v2** |
| **5** | **0** | **v3** |



|  |  |  |  |
| --- | --- | --- | --- |
|  | **length** | **label** | **path** |
| **v1** | **0** | **1** | **--** |
| **v2** | **1** | **1** | **v1** |
| **v3** | **3** | **1** | **v2** |
| **v4** | **4** | **1** | **v2** |
| **v5** | **5** | **1** | **v3** |



v5🡨v3🡨 v2🡨 v1

v4🡨v2🡨v1

**Dijkstra Algorithm**

A weighted directed graph G of n vertices can be described by a matrix D=[d(i,j)]

d(i,j)= weight of directed edge from vertex i to j

d(i,i)=0

d(i,j)=infinity if there is no edge form i to j

Step 1: Assign permanent label 0 to source vertex and temporay label infinity to remaining n-1 vertices.

Step-2: In each iteration another vertex i get permanent label using

Rule:

1. Every vertex j that is not yet permanently labelled gets a new temporary label whose value is given by

min{ old label of j , (old label of i +d(i,j))}

where I is the latest vertex labelled in previous iteration.

1. The smallest value among all the temporary labels is found and it becomes the permanent label of the corresponding vertex. In case of a tie select any one.

Repeat step 1 and 2 until all vertices gets a permanent label.

Time complexity: O(n2)

* Dijkstra algorithm doesn’t not work for negative weights.