SNS COLLEGE OF ENGINEERING

Coimbatore-107

COURSE NAME: ANALYSIS OF ALGORITHM

II YEAR/ IV SEMESTER

UNIT – III

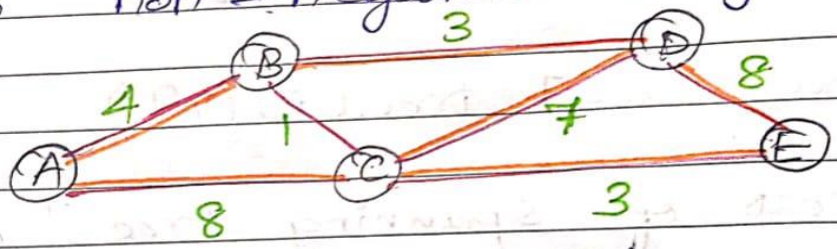
GREEDY TECHNIQUE

Topic

Greedy Technique: Dijkstra’a Algorithm

Unit III - Dijkstra's Algorithm

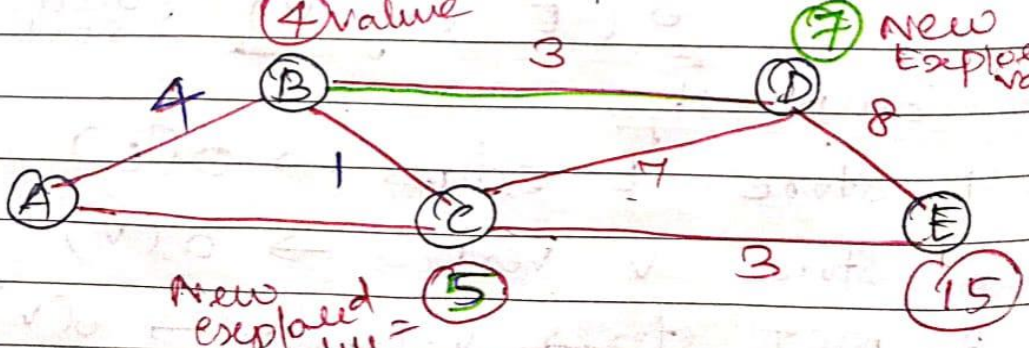
It is a greedy algorithm for finding shortest path from single source vertex to all vertices in weighted graph with non-negative edges.



Source Vertex		Distance with other vertices
1. (A)	A - B	path = 4
	A - C	path = 8
	A - D	path = ∝ (No Direct
	A - E	path = ∝ (No Direct
(B)	B - C	path = 4 + 1 = 5
	B - D	path = 4 + 3 = 7
	B - E	path = ∝

New explored 4 value

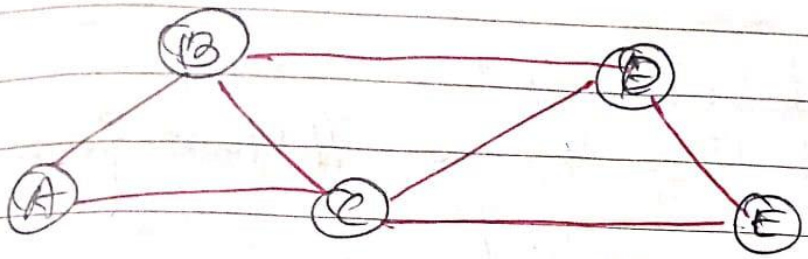
7 New Explored value



New explored value = 5

C-D Path = 5 + 7 = 12

(C) C-E path = 5 + 3 = 8



(D) D-E Path = 7 + 8 = 15

Shortest distance obtained A-E is

Route 1 :⇒ A - B - C - E

4 + 1 + 3 = 8. Not Shortest path

Route 2 :⇒ A - B → D → E Not shortest path.

4 + 3 + 8 = 15

Route 3: A → C → E = 5 + 3 = 8 shortest path.

Route 3: A → B → C → D → E This can't be taken because it have 5 vertices

Route 3 is chosen because it has shortest cost.

Algorithm :

```
Algorithm Dijkstra(Graph[], Source)
for (i=0 to totnodes)
{
dist[i] = Cost[source, i]
path[i] = source // start from source node
}
S[source] = 1
for i=1 to totnodes
{
mindist = ∞
for (j=0 to totnodes-1)
{
if (S[j]=0) then
{
if (dist[j] < mindist)
{
mindist = dist[j]
v1 = j
}
}
S[v1] = 1
for (v2 = 0 to totnodes-1)
{
```

```
if (s[v2] == 0) then
{
if (dist[v1] + cost[v1][v2] < dist[v2])
{
dist[v2] = dist[v1] + cost[v1][v2]
path[v2] = v1
}
}
}
}
}
// v1 is next selected destination
   vertex with shortest distance.
// All these vertices are stored in
   array.
```

Analysis:

Time complexity = $O(V^2)$ depends on Data structure

Space Complexity = $O[V+E]$ for Adjacency List

$O[V^2]$ for Adjacency Matrix.