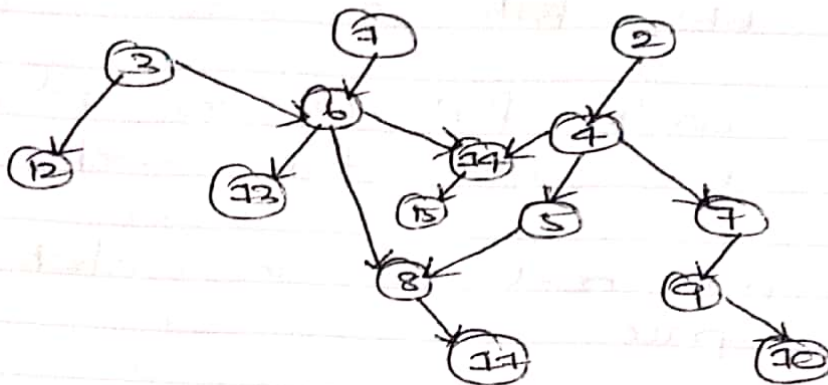


## ASSIGNMENT - 3

### PART - B

B.1) 1) DAG

as course Acyclic graph



2) ALGORITHM Minimum Semesters ( )

{ // pass array & adjacency  
// set list to zero

for (  $i = 0 ; i < N + 1 ; i++$  ) {  
  for (  $j = 0 ; j < N + 1 ; j++$  ) {  
    Adj [ i ] [ j ] = 0 ;

  }

  // Traversal the array of edges.

for (  $i = 0 ; i < N ; i++$  ) {

$x = \text{arr} [ i ] [ 0 ] ;$

$y = \text{arr} [ i ] [ 1 ] ;$

  // update value to 1.

$Adj[x][y] = 1;$

$Adj[y][x] = 1;$

}

// print the minimum value.

B2> GRAPH ALGO (SHORTEST PATH)

1) ALGORITHM SHORT PATH ( )

int v = // get the vertices  
// initialize min heap, dist with  
all vertices.

for (i = 0; i < v; ++i) {

dist[v] = INT\_MAX;

// recursive.

}

Make dist value of src vertex  
as 0, so that it is extracted.

Min Heap  $\rightarrow$   $soa[src] = src;$

// In the following loop, min heap contains all nodes.

while (!is Empty (minheap)) {

// Extract the Min value vertex.

u = minHeapNode  $\rightarrow$  v =

~~At AdjList Node\*~~ = graph  $\rightarrow$  array[u].

while ( )

if (is in Minheap ( ) != INT MAX)

// update dist. value in heap.

} }

// print the calculated shortest path.