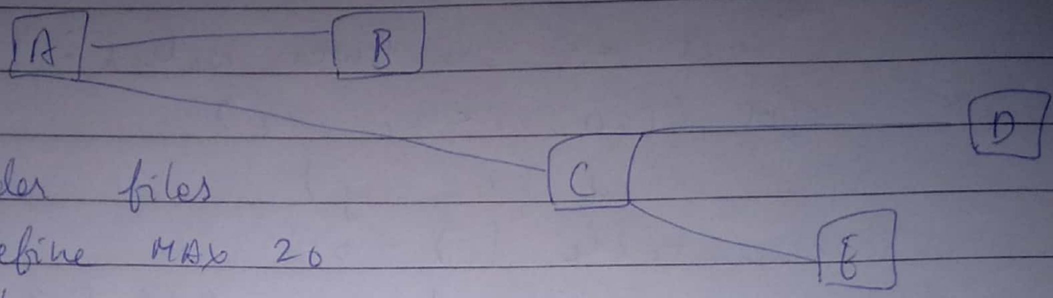
19/11/2020 CN - LAB - 7 1BM18CS0
REVANTH
@[illegible]

Distance vector Algorithm



```
header files
# define MAX 20
 int n
class router
{     char adj_new [ MAX ], adj_old [ MAX ];
 int  new_table [MAX] , old_table [MAX];
 public :
         router ()
 {     for ( int i=0; i < MAX ; i++)
       ~~table~~   ~~old [i] = ( new_table~~
       old_table [i] = new_table [i] = 99;  }
 void    copy ()
{       for (i=0;  i<n;  i++)
    {      adj_old [i] = adj_new [i] ;
           old_table [i] = new_table [i];
          } }
 int  Same ()
 {     for ( int i=0 ; i<n;  i++)
     if ( old_table [i] != new_table [i] )
            return 0 ;
             return 1;
       }
  void      input ( int j )
 {   cout << " Enter 1 if the router is
           adjacent to router " << (char)
     ('A' + j) << " else enter 99 : <<
    for (i=0 ; i<n; i++)
      if ( i!=j)
     cout << (char) ( A +i) << " ";
     cout << "\n Enter matrix: ";
    for (i=0; i<n; i++)
  {         if (i==j)
    new_table [i] = 0;
else           cin >> ~~ta~~ new_table [i];
         adj_new [i] = (char) ('A'+i);
     }

           cout << endl;
       }

    void    display ( )
 {  cout << "\n Destination router: ";
     for (i=0; i<n; i++)
          cout << (char) ('A' + i) << " ";
          cout << "\n outgoing :
      for (i=0; i<n; i++)
         cout << adj_new[i] << " ";
         cout << "\n Count : ";
   for (i=0; i<n; i++)
         cout << ~~table~~. newtable [i] << " "; }
void      build_table ()  {   i=0;    j=0;
     while (i!=n)  {    for (i=j; i<n; i++)
            { r[i]. copy();   r[i]. build(i) );  }
   for (i=0;  i<n;  i++)
            if [!r[i]. equal() {   j=i;   break; }
void main ()
{     cout << " Enter no. of routers
  for (i=0; i<n; i++)  r[i]. input (i);
       build_table();    cout << endl;   getch(); }
```