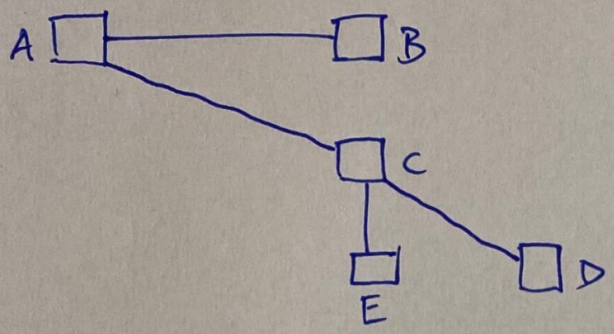
Distance Vector Algorithm

S.R. PUNEETH
1BM18CS087
19/11/20

Topology

5 routers



At every router prior routing table exists

header files

```
#define MAX 10
int n
class router
{
char adj_new[MAX], adj_old[MAX];
int table_new[MAX], table_old[MAX];
public:
router(){
for(int i=0; i<MAX; i++)
 table_old[i] = table_new[i] = 99;
}
void copy()
{
 for(i=0; i<n; i++)
  {
   adj_old[i] = adj_new[i];
   table_old[i] = table_new[i];
  }
}
```

S.R. Puneeth

S.R.TUNEETH
1BM18CS087

```
int equal(){
 for(int i=0;i<n;i++)
 if (table_old[i] != table_new[i]
  adj_new[i] = adj_old[i])
  return 0;
  return 1;
}

void input(int j){
cout<<" Enter \if the corresponding router is adjacent
to router'<<(char)('A'+j)<<" else enter 99 <<endl;
 for (i=0; i<n ;i++)
  if (i!=y)
 cout<<(char)('A'+1)<<" ";
 cout<<"\n Enter matrix";
 for(i=0; i<n ; i++)
  {
  if (i==y)
     table_new[i]=0;
   else
    cin>>table_new[i];
    adj_new[i]=(char)('A'+i);
  }
 cout<<endl;
}
void display()
{
 cout<<"\n Destination routers:";
 for(i=0;i<n;i++) cout<<(char)('A'+i)<<" ";
 cout<<"\n outgoing line;
```

SRI PUNEETH
1BM18CS087

```
for(i=0;i<n;i++) cout<<adj_new[i]<<" ";
 cout<<"\n Hop count: ";
for(i=0; i<n; i++)
 cout<<table_new[i]<<" ";
}
void build(int j)
{
 for(i=0;i<n;i++)
 for(k=0; (i!=j)&&(k<n); k++)
  if(table_old[i]!=99)
  if (table_new[i]+ r[i].table_new[k] < table_new[k]){
   table_new[k]= table_new[i]+ r[i].table_new[k];
   adj_new[k]= (char)('A'+i);
  }}
 }r[10];
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(<"<<MAX<<"): ";
for(i=0; i<n; i++) r[i].input(i);
build_table();
for(i=0; i<n; i++){ cout<<"Router table entries for
router"<<(char)('A'+i)<<" :- ";
r[i].display(); cout<<endl<<endl; } getch(); }
```