HW 9 Yizhan Ao

1. a/ G . $A[1..n, 1..n]$ $\Theta(n^2)$ list $R \Rightarrow G$

Matrix should Use a 2D Array representing Graph

Ininitalize 2D Array size n^2 n vertices arr[n][n]

```
i=0
if(i<n){
   j=0
   if(j<n){
      if(exist edge in i⇒j){
         arr[i][j] =1
      }else{
         arr[i][j] =0
      }
   }else{
   j++
   }
}else{
   i++
}
}
```

b)

```
initialize m=arr[n] with size =n {
        i=0 ,j=0
        if (i<n){
          k= empty linked list
          arr[i] = k
          if (j<n){
            if (edge V exist in [i]->[j]){
               Add to list L
               arr[j]
            }else{
               j++
            }
          else{
           i++
          }
    }
  }
```

2. a) If the proposed shortest path

There are two things needed to check

① CYCLE in the tree

② Root vertex reaching all other

Use DFS check O(m+n).

Based on above we can use pushing root into stack making it to be visited, iratating root through the element, if any thing visited already then, there is a circle we have

b)

```
test (m, adj, visited a)
  visited[m] = true
  for (each n in adj[n]) do
      if (! visited[n] ) then
          test(n, adj, visited)
          if d[n] > d[m] + w(m,n) then
              return false
          end if
      else
          return false
      end if
      return true
  end test
for (i=1 → adj.length)
      visited[i] = false
end for
```

```
| if ( test (V. adj, visited) / then {
      for {j =1 to visited.length }:
          if (! visited [j]) then
            return "No reaching"
          end if
      end for
      else
          return "Tree and Yes"

    else
        return "No"

|   end if
```

3. a) Through edge (a,b) which is shorter

We will make the distance from each vertex to b in case that that vertex [i] if: P[i,a]+s <D[i,b] then D[i,b] = D[i,a] +s. then update the distance between each vertex as much for each vertex [i] and other vertex will be j only if D[i+b] + D[i,j] then D[i,j] = D[i,b] +D[b,j]

b)

Graph "G" : Weight Matrix, "M"

Distance matrix "D" edge (a,b) new edge (a,b)s

```
UpdateDistance( G, M, D, (a,b), s) {
        D[a,b]=s
                  → Vector
        for(i in G.(a,b)){
          if (D[i,a]+s <D[i,b]{
          D[i,b]=D[i,a]+s }
            for(i in G){
            for(j in G){
            if D[i,b]+D[b,j]<D[i,j]{
               D(i,j]=D[i,b]+P[b,j]}
            }
           }
         }
      }
}
```