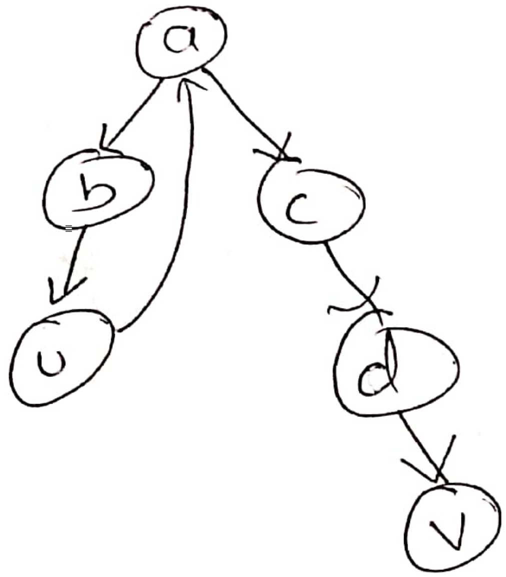
Q1

(a)



from (a)

$d[u] = 2$

$>$

$d[v] = 3$

but u is not descendent of v

(b) ~~Maybe, when graph is~~ : ~~Didn't understand~~

$|E| < |V|^2 \Rightarrow \lg|E| = O(\lg V)$ if there are no multiedges.

(c) No, idea

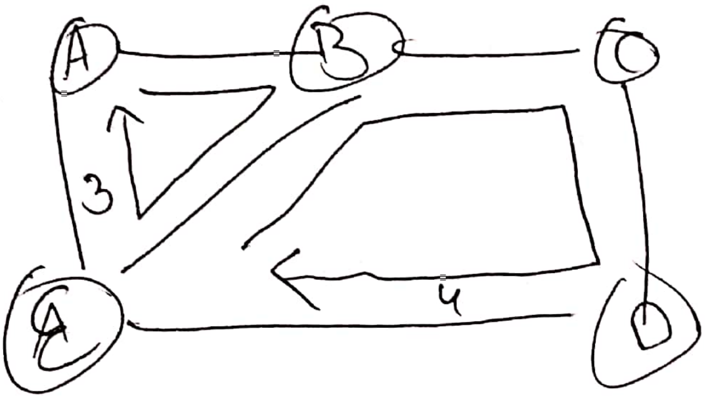
Q2: Store discovery time and finishing time using dfs, u will be ancestor of v if and only if u discovery time is earlier and finishing time is later than that of v.

Q3.

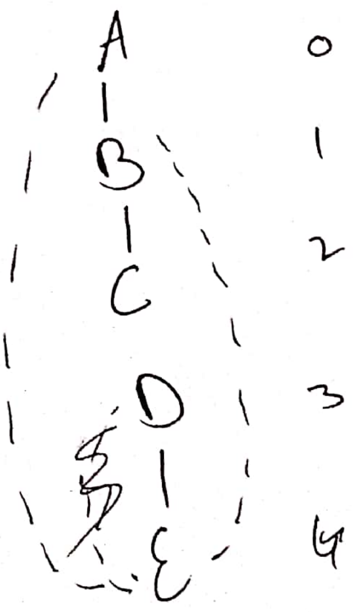
4

5

Counter example



DFS Tree



~~4-3+1=2~~

4 - 1 + 1 = 4 - loop 1

4 - 0 + 1 = 5 - loop2

Q4. For adjacency-list:

```
square (G) :
G1 = G2
For v in G.V :
    For u in adj[v] :
        For w in adj[u] :
            G1.adj[v].add(w)
Remove duplicates in G1
return G1
```

→ This routine will work in $O(VE)$.

Adjacency matrix :

we can convert adjacency matrix in adjacency list in $O(V^2)$ and run the same sub routine

Q6. coinchange (coins [], target) :

```
n = coins.size
dp [0.... target] initialized to ~~zeros~~ +∞
dp [0] = 0
for (int i = 0 ; i <= target ; i++)
    for (int j = 0 ; j <= n ; j++)
        if ( coin[j] <= i )
            dp[i] = min (dp[i], dp[i]-coin[j])

return dp [target]
```