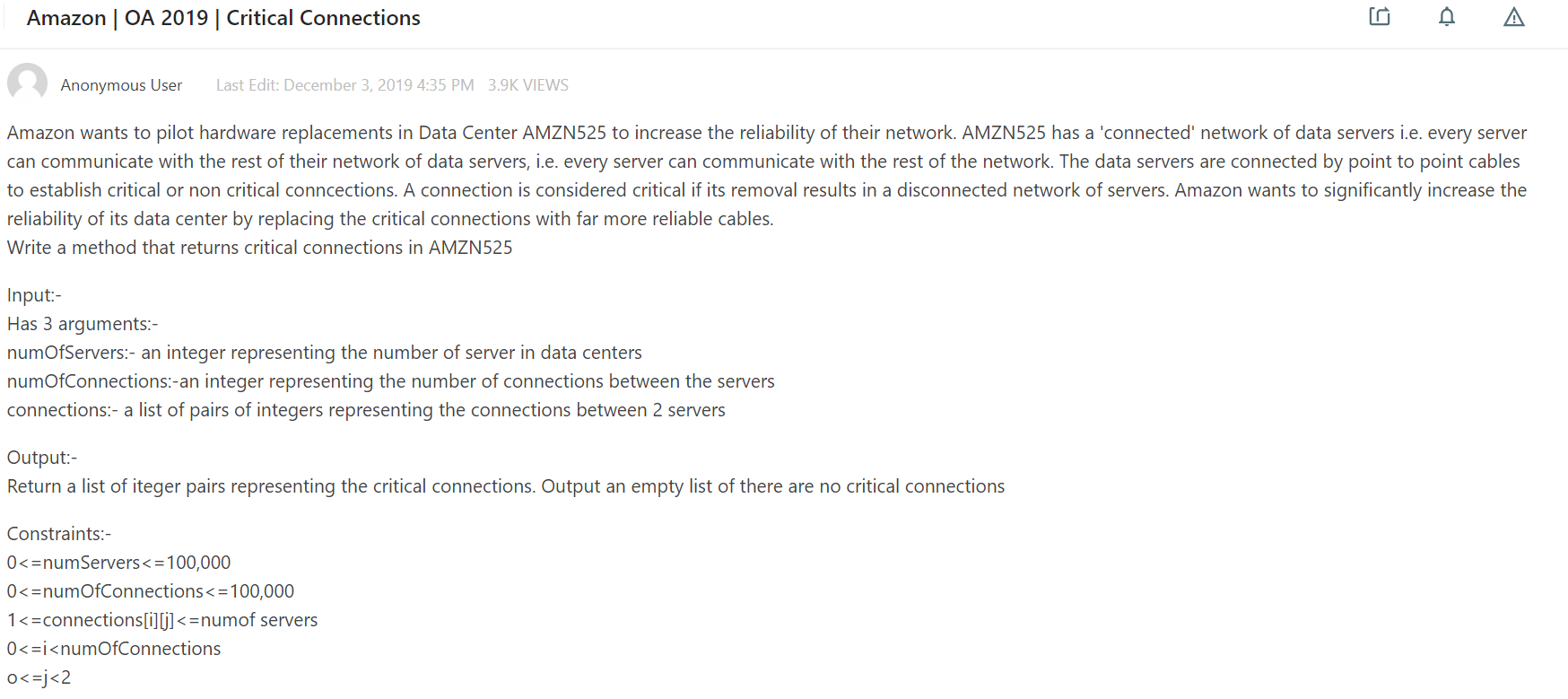
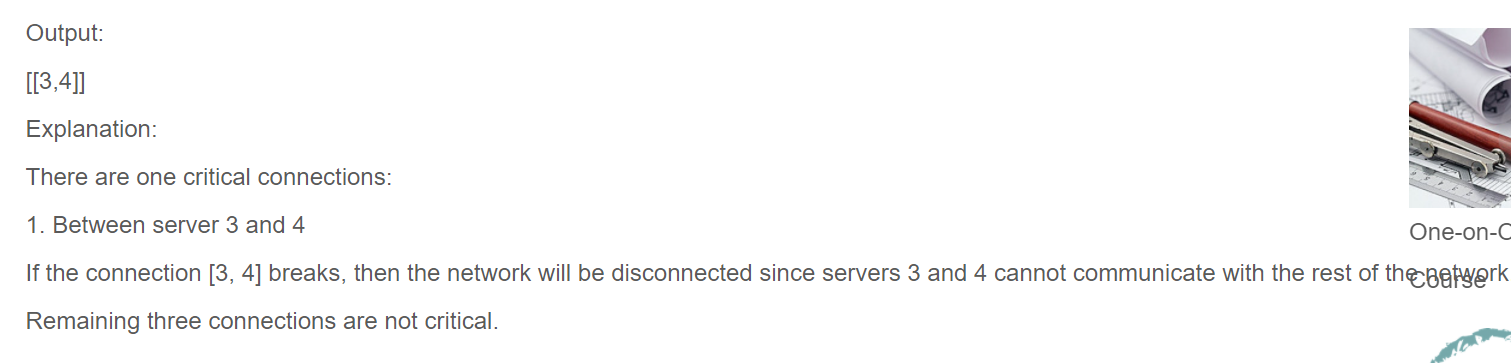
Data Center Critical Connection

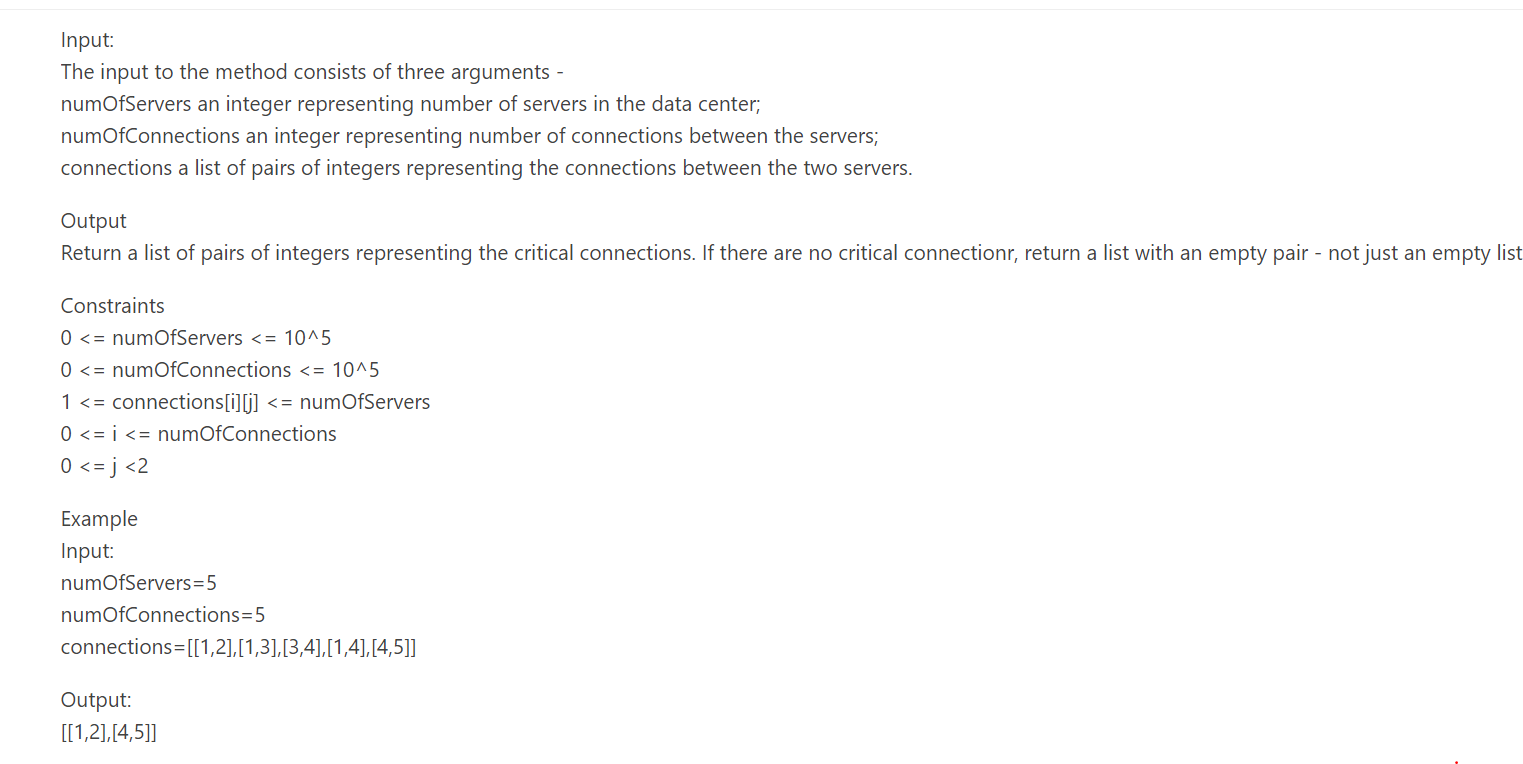
<https://aonecode.com/amazon-online-assessment-data-center-critical-connection>

LEETCODE : https://leetcode.com/discuss/interview-question/868371/amazon-oa-internfte

LEETCODE : https://leetcode.com/discuss/interview-question/393716/amazon-online-assessment-new-grad







void dfs(int u,int& t,vector<vector<int>>& g,vector<int>& par,vector<int>& disc,vector<int>& low,vector<pair<int,int>>& criticalConnections){

disc[u]=low[u]=t++;

for(int v:g[u]){

if(v==par[u])continue;

if(disc[v]==-1){

par[v]=u;

dfs(v,t,g,par,disc,low,criticalConnections);

low[u]=min(low[u],low[v]);

if(disc[u]<low[v])criticalConnections.push\_back({u,v});

}else{

low[u]=min(low[u],disc[v]);

}

}

}

vector<pair<int,int>> findCriticalConnections(int numOfServers,int numOfConnections,vector<pair<int,int>> connections){

vector<vector<int>> g(numOfServers+1);

for(int i=0;i<numOfConnections;i++){

int u=connections[i].first,v=connections[i].second;

g[u].push\_back(v);

g[v].push\_back(u);

}

vector<int> par(numOfServers+1,-1);

vector<int> disc(numOfServers+1,-1);

vector<int> low(numOfServers+1,-1);

int t=1;

vector<pair<int,int>> criticalConnections;

for(int i=1;i<=numOfServers;i++){

if(disc[i]==-1) dfs(i,t,g,par,disc,low,criticalConnections);

}

if(criticalConnections.size()==0)criticalConnections.push\_back({});

return criticalConnections;

}