#Q12 -The complete implementation of disjoint set with path compression and union by rank

class DisjSet:

def \_\_init\_\_(self, n):

self.rank = [1] \* n

self.parent = [i for i in range(n)]

def find(self, x):

if (self.parent[x] != x):

self.parent[x] = self.find(self.parent[x])

return self.parent[x]

def Union(self, x, y):

xset = self.find(x)

yset = self.find(y)

if xset == yset:

return

if self.rank[xset] < self.rank[yset]:

self.parent[xset] = yset

elif self.rank[xset] > self.rank[yset]:

self.parent[yset] = xset

else:

self.parent[yset] = xset

self.rank[xset] = self.rank[xset] + 1

obj = DisjSet(5)

obj.Union(0, 2)

obj.Union(4, 2)

obj.Union(3, 1)

if obj.find(4) == obj.find(0):

print('Yes')

else:

print('No')

if obj.find(1) == obj.find(0):

print('Yes')

else:

print('No')