**Наивная реализация**

color = [i for i in range(8)]  
print(color)  
  
*# индекс массива - номер элемента  
# значение - номер множества*def find(a: int) -> int:  
 return color[a]  
  
  
def union(a: int, b: int) -> None:  
 color\_a = color[a]  
 color\_b = color[b]  
 for i in range(len(color)):  
 if color[i] == color\_b:  
 color[i] = color\_a

**Реализация на списках**

color = [i for i in range(8)]  
color\_list = [[i] for i in range(8)]  
  
  
def find(a: int) -> int:  
 return color[a]  
  
  
def union(a: int, b: int) -> None:  
 color\_a = color[a]  
 color\_b = color[b]  
 if len(color\_list[color\_a]) > len(color\_list[color\_b]):  
 color\_a, color\_b = color\_b, color\_a  
 for x in color\_list[color\_a]:  
 color[x] = color\_b  
color\_list[color\_b] += color\_list[color\_a]  
color\_list[color\_a] = []

union(1, 5)  
union(2, 6)  
union(6, 5)  
union(0, 7)  
print(color)  
print(color\_list)

**Реализация DSU**

p = [-1 for i in range(8)]  
  
  
def find(a: int) -> int:  
 r = a  
 while p[r] != -1:  
 r = p[r]  
 while p[a] != -1:  
 next = p[a]  
 p[a] = r  
 a = next  
 return r  
  
  
def union(a: int, b: int) -> None:  
 p[find(a)] = find(b)  
  
  
union(1, 5)  
union(2, 6)  
union(6, 5)  
union(0, 7)  
find(2)  
print(p)

Острова

color = []  
  
*# индекс массива - номер элемента  
# значение - номер множества*def find(a: int) -> int:  
 return color[a]  
  
  
def union(a: int, b: int) -> None:  
 color\_a = color[a]  
 color\_b = color[b]  
 for i in range(len(color)):  
 if color[i] == color\_b:  
 color[i] = color\_a  
  
  
n, m = map(int, input().split())  
color = [i for i in range(n)]  
for i in range(m):  
 a, b = map(int, input().split())  
 if find(a) != find(b):  
 union(a, b)  
 n -= 1  
 if n == 1:  
 print(i + 1)  
 break

color = []  
color\_list = []  
  
  
def find(a: int) -> int:  
 return color[a]  
  
  
def union(a: int, b: int) -> None:  
 color\_a = color[a]  
 color\_b = color[b]  
 if len(color\_list[color\_a]) > len(color\_list[color\_b]):  
 color\_a, color\_b = color\_b, color\_a  
 for x in color\_list[color\_a]:  
 color[x] = color\_b  
 color\_list[color\_b] += color\_list[color\_a]  
 color\_list[color\_a] = []  
  
n, m = map(int, input().split())  
color = [i for i in range(n)]  
color\_list = [[i] for i in range(n)]  
for i in range(m):  
 a, b = map(int, input().split())  
 if find(a) != find(b):  
 union(a, b)  
 n -= 1  
 if n == 1:  
 print(i + 1)  
 break

p = []  
  
def find(a: int) -> int:  
 r = a  
 while p[r] != -1:  
 r = p[r]  
 while p[a] != -1:  
 next = p[a]  
 p[a] = r  
 a = next  
 return r  
  
  
def union(a: int, b: int) -> None:  
 p[find(a)] = find(b)  
  
  
n, m = map(int, input().split())  
p = [-1 for i in range(n)]  
for i in range(m):  
 a, b = map(int, input().split())  
 if find(a) != find(b):  
 union(a, b)  
 n -= 1  
 if n == 1:  
 print(i + 1)  
 break

DM Algos. Минимальный каркас

p = []  
  
  
def find(a: int) -> int:  
 r = a  
 while p[r] != -1:  
 r = p[r]  
 while p[a] != -1:  
 next = p[a]  
 p[a] = r  
 a = next  
 return r  
  
  
def union(a: int, b: int) -> None:  
 p[find(a)] = find(b)  
  
  
n, m = map(int, input().split())  
p = [-1 for i in range(n + 1)]  
  
edges = []  
for i in range(m):  
 u, v, w = map(int, input().split())  
 edges.append((u, v, w))  
  
edges = sorted(edges, key=lambda x: x[2])  
  
ans = 0  
  
for u, v, w in edges:  
 if find(v) != find(u):  
 union(v, u)  
 ans += w  
  
print(ans)

Разрезание графа  
p = []  
  
  
def find(a: int) -> int:  
 r = a  
 while p[r] != r:  
 r = p[r]  
 while p[a] != a:  
 n = p[a]  
 p[a] = r  
 a = n  
 return r  
  
  
def union(a: int, b: int) -> None:  
 p[find(a)] = find(b)  
  
  
n, m, k = map(int, input().split())  
p = [i for i in range(n + 1)]  
  
for i in range(m):  
 kkk = input()  
  
query = []  
for i in range(k):  
 s, v, u = input().split()  
 query.append((s, int(v), int(u)))  
ans = []  
for i in range(k):  
 qu = query[-1]  
 query.pop()  
 if qu[0] == **"ask"**:  
 if find(qu[1]) == find(qu[2]):  
 ans.append(**"YES"**)  
 else:  
 ans.append(**"NO"**)  
 if qu[0] == **"cut"**:  
 union(qu[1], qu[2])  
  
print(\*ans[::-1], sep=**'**\n**'**)

**Котята**

color = []  
color\_list = []  
  
  
def find(a: int) -> int:  
 return color[a]  
  
  
def union(a: int, b: int) -> None:  
 color\_a = color[a]  
 color\_b = color[b]  
 if len(color\_list[color\_a]) > len(color\_list[color\_b]):  
 color\_a, color\_b = color\_b, color\_a  
 for x in color\_list[color\_a]:  
 color[x] = color\_b  
 color\_list[color\_b] += color\_list[color\_a]  
 color\_list[color\_a] = []  
  
n = int(input())  
color = [i for i in range(n + 1)]  
color\_list = [[i] for i in range(n + 1)]  
for i in range(n - 1):  
 a, b = map(int, input().split())  
 union(a, b)  
for i in range(1, n + 1):  
 if color\_list[i]:  
 print(\*color\_list[i])  
 break